

RESUME MATERI

Double linked list circular

Mata Kuliah : Struktur Data
Dosen Pengampu : Andi Moch Januriana, ST., M.Kom



NAME : Mustopa
NIM : 3337220023
KELAS : C

Program Studi Informatika
Fakultas Teknik
Universitas Sultan Ageng Tirtayasa

APA ITU LINKED LIST ?

- Linked List adalah elemen yang berurutan yang dihubungkan dengan pointer.
- Pointer next / PREV nya akan menunjuk ke diri nya sendiri sehingga berputar (untuk Linked Circular).
- Dapat dibuat selama diperlukan (hingga memori sistem habis).
- Linked List tidak membuang ruang memori (tetapi membutuhkan beberapa memori ekstra untuk pointer).

TIPE LINKED LIST

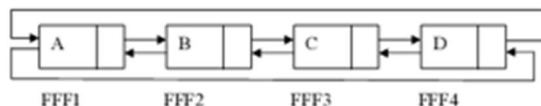
- Single Linked List (Singly Linked List).
- Double Linked List (Doubly Linked List).
- Circular Linked List
- No circular linked List
- Multiple Linked List.

Pada materi ini saya akan meresume materi Double linked list circular atau bisa di singkt DLLC

Pengertian

- Double: artinya field pointer-nya terdiri dari dua buah dan dua arah, yaitu prev dan next
- Circular: artinya pointer next dan prev-nya menunjuk ke dirinya sendiri
- Jika sudah lebih dari satu node, maka pointer prev akan menunjuk ke node sebelumnya, dan pointer next akan menunjuk ke node sesudahnya.

Ilustrasi DLLC



DEKLARASI

```
// deklarasi

struct DataPeminjamanBuku
{
    string nama, nim, NamaBuku;
    int jumlahbuku;

    DataPeminjamanBuku *next;
    DataPeminjamanBuku *prev;
};
```

Disini saya membuat struct dengan nama DataPeminjamanBuku dengan atribut nama,nim DLL

Kemudian membuat variable pointer Next **untuk menunjuk ke node selanjutnya**

Dan variable pointer prev untuk menunjuk ke node sebelumnya

Inisialisasi

```
int main()
{
    DataPeminjamanBuku *databaru;
    databaru = new DataPeminjamanBuku();
    databaru->nama = "cucup";
    databaru->nim = "111";
    databaru->NamaBuku = "bahasa rusia";
    databaru->jumlahbuku = 2;
    databaru->next = databaru;
    databaru->prev = databaru;
```

Disini saya membuat node baru dengan variable pointer databaru

KEYWORD **new** untuk membuat node baru

Kemudian mengisi atribut

- next dan prev nya =databaru **karena node nya masih 1 maka dia akan menunjuk dirinya sendiri** Jika sudah lebih dari satu node, maka pointer prev akan menunjuk ke node sebelumnya, dan pointer next akan menunjuk ke node sesudahnya.

Agar kode terlihat rapih kita masukin kedalam fungsi

```
DataPeminjamanBuku *head, *tail, *bantu, *newNode, *del;
void createDoublelinked(string nama, string nim, string NamaBuku, int jumlahbuku)
{
    head = new DataPeminjamanBuku();
    head->nama = nama;
    head->nim = nim;
    head->jumlahbuku = jumlahbuku;
    head->NamaBuku = NamaBuku;
    head->prev = head;
    head->next = head;
    tail = head;
}
```

Disini sya membuat fungsi dengan parameter semua atribut yang kita tambahkan di dalam struct

Kemudian memanggil semua atribut

Head next dan prev nya = head **karena node nya masih 1 maka dia akan menunjuk dirinya sendiri (circular)**

Tail = **head** dan menjadikan Tail sebagai head (karena node masih 1)

Fungsi menambahkan node dari depan (awal)

```
// tambah depan
void tambahdepan(string nama, string nim, string NamaBuku, int jumlahbuku)
{
    newNode = new DataPeminjamanBuku();
    newNode->nama = nama;
    newNode->nim = nim;
    newNode->jumlahbuku = jumlahbuku;
    newNode->NamaBuku = NamaBuku;

    newNode->prev = tail;
    newNode->next = head;
    head->prev = newNode;
    tail->next = newNode;
    head = newNode;
}
```

Newnode -> prev = tail **artinya node sebelumnya menunjuk ke tail**

newnode->next = head dan node next nya adalah head

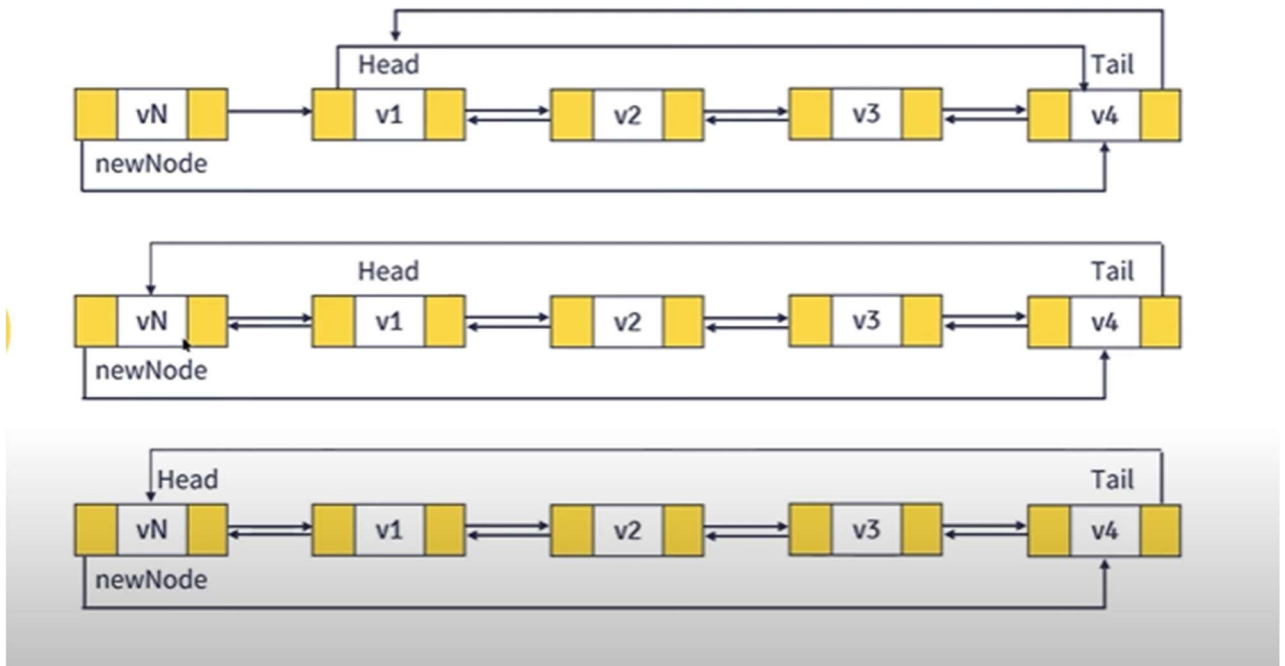
head->prev = newnode node sebelum head adalah newnode

tail->next = newnode **artinya node sesudah tail menunjuk ke newnode**

Head = newnode **dan menjadikan newnode sebagai head**

Ilustrasi

Added at Beginning Node ?



src yt : study with student

Fungsi menampilkan semua node dan atribut

```
// print doublelinkedlist
void printDoublelinkedlist()
{
    bantu = head;

    do
    {
        cout << "Nama :" << bantu->nama << endl;
        cout << "nim :" << bantu->nim << endl;
        cout << "Nama buku :" << bantu->NamaBuku << endl;
        cout << "JumlahBuku :" << bantu->jumlahbuku << endl;
        cout << endl;
        bantu = bantu->next;
    } while (bantu != head);
}
```

kita membuat varibel bantu

dan menjadikan varibel bantu menjadi head

do menampilkan semua atribut

bantu = bantu->next **menjadikan bantu untuk berpindah ke node setelahnya**

while (bantu != head) ketika bantu tidak = head loopinya akan terus berjalan dan menampilkan semua atribut

note : perbedaan do while sama while

do while akan mengerjakan perintah terlebih dahulu kemudian melihat pengkondisian

while akan mengerjakan sesuai kondisi

memanggil fungsi di int main

```
int main()

{
    createDoublelinked("Mustopa ", "333", "mtk", 1);
    tambahdepan("elon", "111", " bahasa indonesia ", 5);
    tambahdepan("JEFBESOS", "444", " bahasa JAWA ", 2);

    printDoublelinkedlist();
}
```

OUTPUT

```
PS C:\Users\USER\Documents\SEMESTER 2\STRUKTUR DATA\DOUBLE LINKEDLIST CIRCULAR> cd "c:\Use
R\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
Nama :JEFBESOS
nim :444
Nama buku : bahasa JAWA
JumlahBuku :2

Nama :elon
nim :111
Nama buku : bahasa indonesia
JumlahBuku :5

Nama :Mustopa
nim :333
Nama buku :mtk
JumlahBuku :1

PS C:\Users\USER\Documents\SEMESTER 2\STRUKTUR DATA\DOUBLE LINKEDLIST CIRCULAR> █
```

SOAL-SOAL

- Buatlah program lengkap dari semua algoritma dan function di atas dalam bentuk menu untuk menambah data, melihat data, dan menghapus data!
- Buatlah function tambahan yang berguna untuk mencari data yang ada dalam linked list baik dengan head maupun head & tail!
- Buatlah function untuk menghapus data tertentu dalam linked list!
- Buatlah function untuk menampilkan data secara terbalik!

- **Buatlah program lengkap dari semua algoritma dan function di atas dalam bentuk menu untuk menambah data, melihat data, dan menghapus data!**

OUTPUT

```
Tampilan Data Peminjaman:
Nama :elon
nim :3333
Nama buku :spaceX
JumlahBuku :6

Nama :rudi
nim :111
Nama buku :mtk
JumlahBuku :1

Nama :Mustopa
nim :3333
Nama buku :admin
JumlahBuku :1

Jumlah elemen dalam linked list: 3
Menu:
1. Tambah Depan Data Peminjaman
2. Hapus Data By Nama Peminjaman
3. Tampilkan Data Peminjaman
4. Edit Data Peminjaman
5. Tampilkan Data Peminjaman Terbalik
6. Keluar
Pilih menu: █
```

KODE LENGKAP :

<https://github.com/mustopa17/Strukdataa>

- Buatlah function tambahan yang berguna untuk mencari data yang ada dalam linked list baik dengan head maupun head & tail!

```
// search data
bool searchByHead(string nama)
{
    bantu = head;

    do
    {
        if (bantu->nama == nama)
            return true;

        bantu = bantu->next;
    } while (bantu != head);

    return false;
}
```

- Buatlah function untuk menghapus data tertentu dalam linked list!

```

// menghapus node tertentu
void removeByname(string nama)
{
    if (head->nama == nama)
    {
        if (head == tail)
        {
            delete head;
            head = NULL;
            tail = NULL;
        }
        else
        {
            DataPeminjamanBuku *temp = head
;
            head = head->next;
            head->prev = tail;
            tail->next = head;
            delete temp;
        }
        cout << "Data berhasil dihapus." <<
endl;
        return;
    }
    bantu = head->next;
    while (bantu != head)
    {
        if (bantu->nama == nama)
        {
            bantu->prev->next = bantu->next
;
            bantu->next->prev = bantu->prev
;
            delete bantu;
            cout <<
            "Data berhasil dihapus." << endl;
            return;
        }
        bantu = bantu->next;
    }
    cout << "Data tidak ditemukan." << endl
;
}

```

- Buatlah function untuk menampilkan data secara terbalik!

```
// PRINT DARI BELAKANG
void printListReverse()
{
    bantu = tail;

    do
    {
        cout << "Nama :" << bantu->nama << endl;
        cout << "nim :" << bantu->nim << endl;
        cout << "Nama buku :" << bantu->NamaBuku << endl;
        cout << "JumlahBuku :" << bantu->jumlahbuku << endl;
        cout << endl;
        bantu = bantu->next;
    } while (bantu != tail);
}
```