

Tugas Pendahuluan Modul 6

STRUKTUR DATA - Ganjil 2024/2025

"06_Double_Linked_List_Bagian_1"

Ketentuan Tugas Pendahuluan

1. Tugas Pendahuluan dikerjakan secara **Individu**.
2. TP ini bersifat **WAJIB**, tidak mengerjakan = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
3. Hanya **MENGUMPULKAN** tetapi **TIDAK MENGERJAKAN** = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
4. Deadline pengumpulan TP Modul 2 adalah Senin, 30 September 2024 pukul 07.30 WIB.
5. **TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN**.
6. **DILARANG PLAGIAT (PLAGIAT = E)**.
7. Kerjakan TP dengan jelas agar dapat dimengerti.
8. Codingan diupload di Github dan upload Laporan di Lab menggunakan format **PDF** dengan ketentuan:
TP_MOD_[XX]_NIM_NAMA.pdf

CP (WA):

- Andini (082243700965)
- Imelda (082135374187)

SELAMAT MENGERJAKAN^^

LAPORAN PRAKTIKUM
PERTEMUAN 6
STRUKTUR DATA



Nama :

Zulfa Mustafa Akhyar Iswahyudi (2311104010)

Dosen :

Yudha Islami Sulistya

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

A. Tujuan

Untuk melatih kompetensi Mahasiswa untuk memperdalam skill pemrograman C++

B. Tools

Codeblocks, VSCode, Github

TUGAS PENDAHULUAN – UNGUIDED

1.) addElementDLL

Program kali ini adalah untuk menambahkan nilai data pada struktur DLL (Double Linked List). Secara struktural dan deklarasi definisi, kebutuhan untuk variabel data dan Node tidak berbeda dengan SLL, namun perbedaannya adalah kita memerlukan pointer *prev* yang tujuannya untuk pengecekan data sebelum Node yang ditentukan.

Sekarang ada method baru untuk menambahkan data pada awal DLL, yaitu **'insertFirst'**. Isi method ini sama seperti **'insertLast'**, bedanya ada di inisialisasi set default nilai data dalam list-nya. Kalo pada **'insertFirst'**, inisialisasi nilainya dimulai dari data pertama dalam Node DLL.

Pada pengoperasian utama pada program, kita bisa memilih 4 opsi menu yang disediakan. Kita bisa menginput data dari depan DLL pada menu 1, menginput data dari belakang DLL pada menu 2, menampilkan keseluruhan nilai DLL, dan exit untuk keluar dari program.

Oh iya, jangan lupa untuk memasukkan method yang sesuai dengan menu agar program dapat berjalan secara terstruktur.

```

EXPLORE
STD_ZULFA_MUSTANA_ABDINNA
> [v] vscode
> [v] 01_pengenalan_CPP_3...
> [v] 02_pengenalan_CPP_3...
> [v] 03_Abstract_Data_Type...
> [v] 04_Single_Linked_List...
> [v] 05_Single_Linked_List...
> [v] PRAKTIKUM_5
> [v] vscode
> [v] KodingsBar...
> [v] Latihan_Ungu...
> [v] TugasPendak...
> [v] output
C- bubbleSort... U
C- newAddElem... U
C- searchElem... U
[?] TP_MOD_5_231... U
[?] TP_MOD_5_231... U
> [v] 06_Double_Link...
> [v] PRAKTIKUM_6
> [v] KodingsBar...
> [v] Latihan_Ungu...
> [v] TugasPendak...
> [v] output
C- addElements... U
C- deleteElem... U
C- displayAcD... U
[?] -SOM NURRA... U
[?] BLOM NURRA... U

06_Double_Linked_List_Bagian_1 > PRAKTIKUM_6 > TugasPendakhan > C- addElementsDLL.cpp > [v] main()
1 // Menambahkan elemen di Awal dan Akhir DLL
2 #include <iostream>
3 using namespace std;
4
5 struct Node
6 {
7     int data;
8     Node *prev;
9     Node *next;
10 };
11
12 Node *head = nullptr;
13 Node *tail = nullptr;
14
15 Code Gen: Options | Test this function
16 void insertFirst_2311104010(int data)
17 {
18     Node *newNode = new Node();
19     newNode->data = data;
20     newNode->prev = nullptr;
21     newNode->next = head;
22
23     if (head != nullptr)
24     {
25         head->prev = newNode;
26     }
27     head = newNode;
28
29     if (tail == nullptr)
30     {
31         tail = head;
32     }
33 }
34
35 Code Gen: Options | Test this function
36 void insertLast_2311104010(int data)
37 {
38     Node *newNode = new Node();
39     newNode->data = data;
40     newNode->next = nullptr;
41     newNode->prev = tail;
42
43     if (tail != nullptr)
44     {
45         tail->next = newNode;
46     }
47     tail = newNode;
48
49     if (head == nullptr)
50     {
51         head = tail;
52     }
53 }
54
55 Code Gen: Options | Test this function
56 void displayForward_2311104010()
57 {
58     Node *temp = head;
59     while (temp != nullptr)
60     {
61         cout << temp->data;
62         if (temp->next != nullptr)
63             cout << " <-> ";
64         temp = temp->next;
65     }
66     cout << endl;
67 }
68
69 Code Gen: Options | Test this function
70 int main()
71 {
72     int choice, data;
73     do
74     {
75         cout << "\nMenu:\n";
76         cout << "1. Insert depan\n";
77         cout << "2. Insert belakang\n";
78         cout << "3. Display\n";
79         cout << "4. Exit\n";
80         cout << "Apa pilihannya?: ";
81         cin >> choice;
82
83         switch (choice)
84         {
85             case 1:
86                 cout << "Masukan data dari depan list data: ";
87                 cin >> data;
88                 insertFirst_2311104010(data);
89                 break;
90             case 2:
91                 cout << "Masukan data dari belakang list data: ";
92                 cin >> data;
93                 insertLast_2311104010(data);
94                 break;
95             case 3:
96                 cout << "Ini keseluruhan data saat ini: ";
97                 displayForward_2311104010();
98                 break;
99             case 4:
100                 cout << "Yak keluar\n";
101                 break;
102             default:
103                 cout << "Tak ada opsinya.\n";
104         }
105     } while (choice != 4);
106
107     return 0;
108 }

```

Output :

```
ATA\STD_ZULFA_MUSTAFA_AKHYAR_ISWAHYUDI_2311104  
PS C:\Users\HUAWEI\OneDrive\Documents\ALL_ITTF  
.\'addElementDLL.exe'
```

Menu:

1. Insert depan
2. Insert belakang
3. Display
4. Exit

Apa pilihanmu?: 1

Masukan data dari depan list data: 5

Menu:

1. Insert depan
2. Insert belakang
3. Display
4. Exit

Apa pilihanmu?: 1

Masukan data dari depan list data: 2

Menu:

1. Insert depan
2. Insert belakang
3. Display
4. Exit

Apa pilihanmu?: 3

Ini keseluruhan data saat ini: 2 <-> 5

Menu:

1. Insert depan
2. Insert belakang
3. Display
4. Exit

Apa pilihanmu?: 2

Masukan data dari depan list data: 10

Menu:

1. Insert depan
2. Insert belakang
3. Display
4. Exit

Apa pilihanmu?: 2

Masukan data dari depan list data: 7

```
Menu:
1. Insert depan
2. Insert belakang
3. Display
4. Exit
Apa pilihanmu?: 3
Ini keseluruhan data saat ini: 2 <-> 5 <-> 10 <-> 7

Menu:
1. Insert depan
2. Insert belakang
3. Display
4. Exit
Apa pilihanmu?: 4
Yok keluar
```

2.) deleteElementDLL

Program kali ini sama, yaitu DLL.

Hanya ada dua method tambahan disini, yaitu **'deleteFirst'** dan **'deleteLast'**. Keduanya memiliki isi yang sama, yaitu sintaks untuk mendeteksi data yang akan dihapus, hanya saja jika **'deleteFirst'** akan menghapus data pertama dalam DLL. Sementara **'deleteLast'** akan menghapus data terakhir dalam DLL.

Pada operasi utamanya kita disediakan menu yang sama seperti file diatas, namun bedanya kita tidak menginput data pada bagian depan atau belakang DLL, melainkan kita akan menghapus data inputan pada menu pertama baik itu dibelakang maupun didepan DLL.

[illegible]

Output :

```
ATA\STD_ZULFA_MUSTAFA_AKHYAR_ISWAHYUDI_2311104010\
PS C:\Users\HUAWEI\OneDrive\Documents\ALL_ITTP_SEM
.\'deleteElementDLL.exe'
```

Menu:

1. Masukan data diakhir DLL
2. Hapus data pertama
3. Hapus data terakhir
4. Tampilkan keseluruhan
5. Keluar

Masukan opsinya: 1

Masukan data diakhir DLL: 10

Menu:

1. Masukan data diakhir DLL
2. Hapus data pertama
3. Hapus data terakhir
4. Tampilkan keseluruhan
5. Keluar

Masukan opsinya: 1

Masukan data diakhir DLL: 8

Menu:

1. Masukan data diakhir DLL
2. Hapus data pertama
3. Hapus data terakhir
4. Tampilkan keseluruhan
5. Keluar

Masukan opsinya: 1

Masukan data diakhir DLL: 6

Menu:

1. Masukan data diakhir DLL
2. Hapus data pertama
3. Hapus data terakhir
4. Tampilkan keseluruhan
5. Keluar

Masukan opsinya: 4

Ini keseluruhan datanya saat ini: 10 <-> 8 <-> 6


```
Menu:
1. Masukan data diakhir DLL
2. Hapus data pertama
3. Hapus data terakhir
4. Tampilkan keseluruhan
5. Keluar
Masukan opsinya: 2
Hapus data pertama.

Menu:
1. Masukan data diakhir DLL
2. Hapus data pertama
3. Hapus data terakhir
4. Tampilkan keseluruhan
5. Keluar
Masukan opsinya: 3
Hapus data terakhir.

Menu:
1. Masukan data diakhir DLL
2. Hapus data pertama
3. Hapus data terakhir
4. Tampilkan keseluruhan
5. Keluar
Masukan opsinya: 4
Ini keseluruhan datanya saat ini: 8

Menu:
1. Masukan data diakhir DLL
2. Hapus data pertama
3. Hapus data terakhir
4. Tampilkan keseluruhan
5. Keluar
Masukan opsinya: 5
Lagi keluar
```

3.) displayAscDescDLL

Kita hanya punya dua method berbeda, yaitu **'displayAscending'** dan **'displayDescending'**. Keduanya sama-sama memiliki perulangan untuk mendeteksi seluruh data DLL lalu menampilkannya pada Output. Bedanya hanya pada **'displayAscending'** akan menampilkan urutan seluruh data dari depan. Sedangkan **'displayDescending'** akan menampilkan urutan seluruh data dari belakang.

Pada operasi programnya, seperti biasa ada menu untuk menginput data dan keluar dari program. Dua menu lainnya berfungsi untuk menampilkan hasil seluruh inputan data pada DLL sesuai dengan method-nya masing-masing.

```

01_Double_Linked_List_Bagian_1 > PRAKTIKUM_5 > TugasPendekutan > displayAscDesc.cpp > insertasi_2311104010
// Anampilkan elemen dari Depan ke Belakang dan Sebaliknya
#include <iostream>
using namespace std;

struct Node
{
    int data;
    Node *prev;
    Node *next;
};

Node *head = nullptr;
Node *tail = nullptr;

// Insertasi Depan
void insertasi_2311104010(int data)
{
    Node *newNode = new Node();
    newNode->data = data;
    newNode->next = nullptr;
    newNode->prev = tail;
    newNode->prev = tail;

    if (tail != nullptr)
    {
        tail->next = newNode;
        tail = newNode;
    }
    else if (head == nullptr)
    {
        head = tail;
    }
}

// Tampil dari depan data
// Goto Goto Output / Terima Terima
void displayAscending_2311104010()
{
    Node *temp = head;
    while (temp != nullptr)
    {
        cout << temp->data;
        if (temp->next != nullptr)
            cout << " <-> ";
        temp = temp->next;
    }
    cout << endl;
}

// Tampil dari belakang data
// Goto Goto Output / Terima Terima
void displayDescending_2311104010()
{
    Node *temp = tail;
    while (temp != nullptr)
    {
        cout << temp->data;
        if (temp->prev != nullptr)
            cout << " <-> ";
        temp = temp->prev;
    }
    cout << endl;
}

// Goto Goto Output / Terima Terima
int main()
{
    int choice, data;
    do
    {
        cout << "\nMenu:\n";
        cout << "1. Masukkan data diakhir List\n";
        cout << "2. Tampilkan seluruh data urut dari belakang\n";
        cout << "3. Tampilkan seluruh data urut dari depan\n";
        cout << "4. Exit\n";
        cout << "Enter your choice: ";
        cin >> choice;

        switch (choice)
        {
            case 1:
                cout << "Masukkan data diakhir List: ";
                cin >> data;
                insertasi_2311104010(data);
                break;
            case 2:
                cout << "Tampilkan seluruh data urut dari belakang: ";
                displayDescending_2311104010();
                break;
            case 3:
                cout << "Tampilkan seluruh data urut dari depan: ";
                displayAscending_2311104010();
                break;
            case 4:
                cout << "Exiting\n";
                break;
            default:
                cout << "Invalid choice, please try again.\n";
        }
    } while (choice != 4);

    return 0;
}

```

Output :

Menu:

1. Masukkan data diakhir List
2. Tampilkan seluruh data urut dari belakang
3. Tampilkan seluruh data urut dari depan
4. Exit

Enter your choice: 1

Masukan data diakhir List: 50

Menu:

1. Masukkan data diakhir List
2. Tampilkan seluruh data urut dari belakang
3. Tampilkan seluruh data urut dari depan
4. Exit

Enter your choice: 1

Masukan data diakhir List: 60

Menu:

1. Masukkan data diakhir List
2. Tampilkan seluruh data urut dari belakang
3. Tampilkan seluruh data urut dari depan
4. Exit

Enter your choice: 1

Masukan data diakhir List: 40

Menu:

1. Masukkan data diakhir List
2. Tampilkan seluruh data urut dari belakang
3. Tampilkan seluruh data urut dari depan
4. Exit

Enter your choice: 2

Tampilkan seluruh data urut dari belakang: 50 <-> 60 <-> 40

Menu:

1. Masukkan data diakhir List
2. Tampilkan seluruh data urut dari belakang
3. Tampilkan seluruh data urut dari depan
4. Exit

Enter your choice: 3

Tampilkan seluruh data urut dari depan: 40 <-> 60 <-> 50

Menu:

1. Masukkan data diakhir List
2. Tampilkan seluruh data urut dari belakang
3. Tampilkan seluruh data urut dari depan
4. Exit

Enter your choice: 4

Exiting

LATIHAN - UNGUIDED

1.) Mod_6

Kali ini programnya lumayan rumit. Deklarasi data dalam constructor harus terdiri dari IDBuku, judul, author, dan pointer prev / next pada constructor-nya.

Untuk method pertama ada 'addBook'. Isinya adalah memperbarui variabel-variabel sebelumnya menjadi objek yang menampung nilai sesuai tipe datanya masing-masing.

Selanjutnya ada dua method untuk menampilkan keseluruhan data DLL yaitu 'DisFirst' dan 'DisLast'. Keduanya memiliki kegunaan yang sama yaitu menampilkan data judul, author, dan IDBuku namun berbeda urutannya yaitu depan dan belakang

Pada program utamanya, kita cukup menuliskan data IDBuku, judul, dan author-nya menggunakan method addBook pada Source Code. Kemudian untuk menampilkan outputnya gunakan dua method Display tadi.

```

01 //include <iostream>
02 //include <string>
03
04 using namespace std;
05
06 struct Buku
07 {
08     int IDBuku;
09     string Judul;
10     string author;
11     Buku *prev;
12     Buku *next;
13 };
14
15 // Fungsi untuk menambahkan buku di akhir linked list
16 // Goto Gen Options | Test the function
17 void addBook1_2311104010(Buku *head, int id, string judul, string author)
18 {
19     Buku *newBuku = new Buku;
20     newBuku->IDBuku = id;
21     newBuku->Judul = judul;
22     newBuku->author = author;
23     newBuku->next = nullptr;
24
25     if (head == nullptr)
26     {
27         newBuku->prev = nullptr;
28         head = newBuku;
29     }
30     else
31     {
32         Buku *temp = head;
33         while (temp->next != nullptr)
34         {
35             temp = temp->next;
36         }
37         temp->next = newBuku;
38         newBuku->prev = temp;
39     }
40 }
41
42 // Fungsi untuk menampilkan daftar buku dari awal ke akhir
43 // Goto Gen Options | Test the function
44 void DisFirst_2311104010(Buku *head)
45 {
46     Buku *temp = head;
47     while (temp != nullptr)
48     {
49         cout << "ID Buku: " << temp->IDBuku << ", Judul: " << temp->Judul
50             << ", author: " << temp->author << endl;
51         temp = temp->next;
52     }
53 }
54
55 // Fungsi untuk menampilkan daftar buku dari akhir ke awal
56 // Goto Gen Options | Test the function
57 void DisLast_2311104010(Buku *head)
58 {
59     if (head == nullptr)
60     {
61         return;
62     }
63     Buku *temp = head;
64     while (temp->next != nullptr)
65     {
66         temp = temp->next;
67     }
68     while (temp != nullptr)
69     {
70         cout << "ID Buku: " << temp->IDBuku << ", Judul: " << temp->Judul
71             << ", author: " << temp->author << endl;
72         temp = temp->prev;
73     }
74 }
75
76 // Goto Gen Options | Test the function
77 int main()
78 {
79     Buku *head = nullptr;
80
81     addBook1_2311104010(head, 101, "Unwritten Wanderer Saga I : Resilience Era's of Fathomless Warrior", "Bangjoule");
82     addBook1_2311104010(head, 102, "Unwritten Wanderer Saga II : Eastern Daylight Has Surken To The Naraka", "Bangjoule");
83     addBook1_2311104010(head, 103, "Unwritten Wanderer Saga III : Across The Old Covenant, In The Distance Luminaries Ocean", "Bangjoule");
84
85     cout << "Daftar buku dari awal ke akhir: " << endl;
86     DisFirst_2311104010(head);
87
88     cout << "Daftar buku dari akhir ke awal: " << endl;
89     DisLast_2311104010(head);
90
91     return 0;
92 }

```

Output :

```
.\'mod6.exe'  
Daftar buku dari awal ke akhir:  
ID Buku: 101, Judul: Unwritten Wanderer Saga I : Resilience Era's of Fathomless Warrior, author: Bangjoule  
ID Buku: 102, Judul: Unwritten Wanderer Saga II : Eastern Daylight Has Sunken To The Naraka, author: Bangjoule  
ID Buku: 103, Judul: Unwritten Wanderer Saga III : Across The Olde Covenant, In The Distance Luminaries Ocean, author: Bangjoule  
Daftar buku dari akhir ke awal:  
ID Buku: 103, Judul: Unwritten Wanderer Saga III : Across The Olde Covenant, In The Distance Luminaries Ocean, author: Bangjoule  
ID Buku: 102, Judul: Unwritten Wanderer Saga II : Eastern Daylight Has Sunken To The Naraka, author: Bangjoule  
ID Buku: 101, Judul: Unwritten Wanderer Saga I : Resilience Era's of Fathomless Warrior, author: Bangjoule
```

SOURCE CODE WAKTU DI KELAS – KODINGAN ASPRAK

```

DLL.cpp

#include <iostream>
#include <string>
using namespace std;

class Node
{
public:
    int data;
    Node *prev;
    Node *next;
};

class DoublyLinkedList
{
public:
    Node *head;
    Node *tail;

    // Constructor untuk inisialisasi head dan tail
    DoublyLinkedList() : head(nullptr), tail(nullptr) {}

    // Destructor untuk membersihkan semua elemen dari memori
    ~DoublyLinkedList()
    {
        deleteAll();
    }

    // Fungsi untuk menambahkan elemen di depan list (push)
    void push(int data)
    {
        Node *newNode = new Node;
        newNode->data = data;
        newNode->prev = nullptr;
        newNode->next = head;

        if (head != nullptr)
        {
            head->prev = newNode;
        }
        else
        {
            tail = newNode; // Jika list kosong, tail juga mengarah ke node baru
        }
        head = newNode;
    }

    // Fungsi untuk menghapus elemen dari depan list (pop)
    void pop()
    {
        if (head == nullptr)
        {
            cout << "List is empty!" << endl;
            return; // Jika list kosong
        }
        Node *temp = head;
        head = head->next;
        if (head == nullptr)
        {
            head->prev = nullptr;
        }
        else
        {
            tail = nullptr; // Jika hanya satu elemen di list
        }
        delete temp; // Hapus elemen
    }

    // Fungsi untuk mengupdate data di list
    bool update(int oldData, int newData)
    {
        Node *current = head;
        while (current != nullptr)
        {
            if (current->data == oldData)
            {
                current->data = newData;
                return true; // Jika data ditemukan dan diupdate
            }
            current = current->next;
        }
        return false; // Jika data tidak ditemukan
    }

    // Fungsi untuk menghapus semua elemen di list
    void deleteAll()
    {
        Node *current = head;
        while (current != nullptr)
        {
            Node *temp = current;
            current = current->next;
            delete temp;
        }
        head = nullptr;
        tail = nullptr;
    }

    // Fungsi untuk menampilkan semua elemen di list
    void display()
    {
        if (head == nullptr)
        {
            cout << "List is empty!" << endl;
            return;
        }
        Node *current = head;
        while (current != nullptr)
        {
            cout << current->data << " ";
            current = current->next;
        }
        cout << endl;
    }
};

// Fungsi untuk memvalidasi input integer
bool validateInput(int &input)
{
    if (cin.get() != '\n')
    {
        cin.clear();
        cout << "Invalid input. Please enter a valid number." << endl;
        return false;
    }
    return true;
}

int main()
{
    DoublyLinkedList list;
    while (true)
    {
        cout << "Menu:" << endl;
        cout << "1. Add data" << endl;
        cout << "2. Delete data" << endl;
        cout << "3. Update data" << endl;
        cout << "4. Clear data" << endl;
        cout << "5. Display data" << endl;
        cout << "6. Exit" << endl;

        int choice;
        cout << "Enter your choice: ";
        cin >> choice;

        if (!validateInput(choice))
            continue;

        switch (choice)
        {
            case 1:
            {
                int data;
                cout << "Enter data to add: ";
                cin >> data;

                if (validateInput(data))
                {
                    list.push(data);
                }
                break;
            }
            case 2:
            {
                list.pop();
                break;
            }
            case 3:
            {
                int oldData, newData;
                cout << "Enter old data: ";
                cin >> oldData;

                if (!validateInput(oldData))
                    continue;

                cout << "Enter new data: ";
                cin >> newData;

                if (!validateInput(newData))
                    continue;

                bool updated = list.update(oldData, newData);
                if (!updated)
                {
                    cout << "Data not found" << endl;
                }
                break;
            }
            case 4:
            {
                list.deleteAll();
                cout << "All data cleared." << endl;
                break;
            }
            case 5:
            {
                list.display();
                break;
            }
            case 6:
            {
                cout << "Exiting the program." << endl;
                return 0;
            }
            default:
            {
                cout << "Invalid choice. Please try again." << endl;
                break;
            }
        }
    }
}

```

Semoga Selalu diberi kemudahan^^