LAPORAN PRAKTIKUM
STRUKTUR DATA

MODUL 7
STACK

Disusun Oleh :

NAMA : Faris Walid Awwal Aidi

NIM : 103112430133

Dosen

FAHRUDIN MUKTI WIBOWO

PROGRAM STUDI STRUKTUR DATA

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2025

A. Dasar Teori

Stack (Tumpukan) merupakan salah satu bentuk struktur data yang menerapkan prinsip operasi seperti tumpukan. Prinsip utama yang digunakan dalam stack adalah LIFO (Last In First Out). Artinya, elemen yang terakhir kali dimasukkan adalah elemen yang pertama kali dapat diambil, atau dengan kata lain, elemen yang bisa diambil terlebih dahulu hanyalah elemen yang berada di posisi paling atas. Komponen utama dalam stack yang berfungsi sebagai penanda sekaligus pengakses data adalah "Top". Akses pada stack memang terbatas, yaitu hanya bisa dilakukan pada awal stack saja (elemen Top).

Terdapat dua operasi utama yang fundamental dalam stack, yaitu operasi Push dan Pop. Push adalah operasi untuk menyisipkan atau menambahkan elemen baru ke dalam tumpukan. Operasi ini identik dengan fungsi insert first pada linked list biasa. Sementara itu, Pop adalah operasi untuk mengambil atau mengeluarkan data dari tumpukan. Operasi ini mirip dengan delete first pada list linear, karena elemen yang paling pertama kali diakses (untuk diambil) adalah elemen paling atas atau paling awal. Stack dapat diimplementasikan menggunakan dua cara, yaitu representasi pointer (menggunakan singly linked list) dan representasi tabel (menggunakan array).

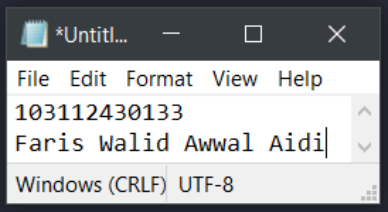
.

B. Guided (berisi screenshot source code & output program disertai penjelasannya)

Guided

guided > stack.cpp > ...

```cpp
#include <iostream>
using namespace std;

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

bool isEmpty(Node *top)
{
    return top == nullptr;
}

void push(Node *&top, int data)
{
    Node *newNode = new Node();
    newNode->data = data;
    newNode->next = top;
    top = newNode;
}

int pop(Node *&top)
{
    if (isEmpty(top))
    {
        cout << "Stack kosong, tidak dapat pop!" << endl;
        return 0;
    }

    int poppedData = top->data;
    Node *temp = top;
    top = top->next;

    delete temp;
    return poppedData;
}
```



```cpp
void show(Node *top)
{
    if (isEmpty(top))
    {
        cout << "Stack Kosong." << endl;
        return;
    }

    cout << "TOP -> ";
    Node *temp = top;

    while (temp != nullptr)
    {
        cout << temp->data << " -> ";
        temp = temp->next;
    }

    cout << "NULL" << endl;
}

int main()
{
    Node *stack = nullptr;

    push(stack, 10);
    push(stack, 20);
    push(stack, 30);

    cout << "Menampilkan isi stack:" << endl;
    show(stack);

    cout << "Pop: " << pop(stack) << endl;

    cout << "Menampilkan sisa stack:" << endl;
    show(stack);

    return 0;
}
```

*Untitl...
File Edit Format View Help
103112430133
Faris Walid Awwal Aidi
Windows (CRLF) UTF-8

Screenshots Output

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

PS D:\C++\modul7_30okt25> cd .\guided\
PS D:\C++\modul7_30okt25\guided> .\stack.exe
Menampilkan isi stack:
TOP -> 30 -> 20 -> 10 -> NULL
Pop: 30
Menampilkan sisa stack:
TOP -> 20 -> 10 -> NULL
PS D:\C++\modul7_30okt25\guided>
```

*Untitl...
File Edit Format View Help
103112430133
Faris Walid Awwal Aidi
Windows (CRLF) UTF-8

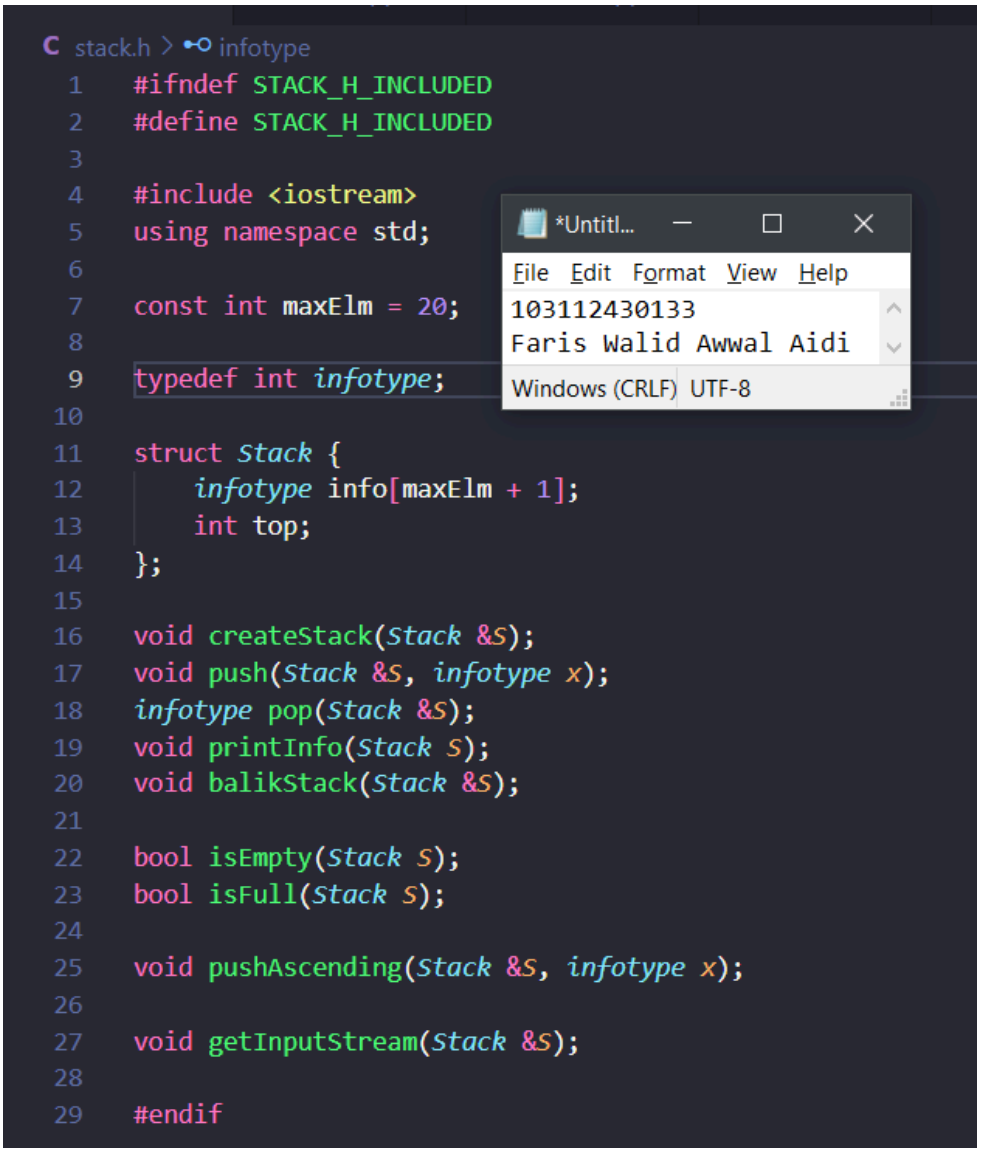
Deskripsi:

Program ini adalah implementasi struktur data Stack (Tumpukan) sederhana menggunakan linked list di C++. Program ini menggunakan struct Node untuk menyimpan data (integer) dan pointer next yang menunjuk ke elemen di bawahnya. Sebuah pointer bernama stack (dalam main) bertindak sebagai top untuk melacak elemen teratas. Fungsi utamanya adalah push untuk menambahkan elemen baru ke atas tumpukan (proses insert first) dan pop untuk mengambil sekaligus menghapus elemen teratas (proses delete first). Program ini juga dilengkapi fungsi isEmpty untuk mengecek kekosongan stack dan fungsi show untuk mencetak semua elemen. Fungsi main hanya mendemonstrasikan cara kerja dasar dengan memasukkan tiga data, menampilkannya, mengambil satu data, lalu menampilkannya lagi.

C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

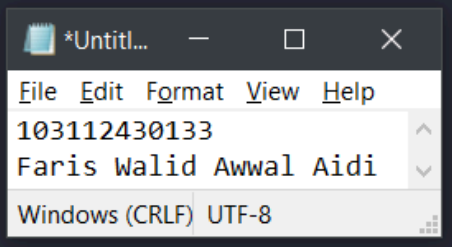
- stack.h



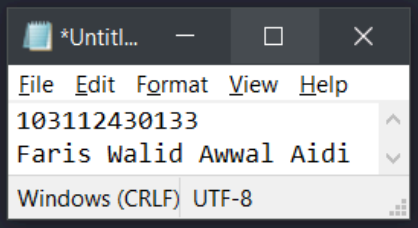
- stack.cpp

stack.cpp > isEmpty(Stack)

```cpp
#include "stack.h"

void createStack(Stack &S) {
    S.top = 0;
}

bool isEmpty(Stack S) {
    return S.top == 0;
}

bool isFull(Stack S) {
    return S.top == maxElm;
}

void push(Stack &S, infotype x) {
    if (!isFull(S)) {
        S.top++;
        S.info[S.top] = x;
    }
}

infotype pop(Stack &S) {
    infotype x;
    if (!isEmpty(S)) {
        x = S.info[S.top];
        S.top--;
    }
    return x;
}

void printInfo(Stack S) {
    cout << "[TOP] ";
    if (!isEmpty(S)) {
        for (int i = S.top; i >= 1; i--) {
            cout << S.info[i] << " ";
        }
    }
    cout << endl;
}
```



```
}

void balikStack(Stack &S) {
    Stack tempStack;
    createStack(tempStack);

    while (!isEmpty(S)) {
        push(tempStack, pop(S));
    }

    S = tempStack;
}

void pushAscending(Stack &S, infotype x) {
    Stack tempStack;
    createStack(tempStack);

    while (!isEmpty(S) && S.info[S.top] > x) {
        push(tempStack, pop(S));
    }

    push(S, x);

    while (!isEmpty(tempStack)) {
        push(S, pop(tempStack));
    }
}

void getInputStream(Stack &S) {
    char c;
    c = cin.get();

    while (c != '\n') {
        infotype x = c - '0';
        push(S, x);
        c = cin.get();
    }
}
```



- main.cpp untuk nomor 1

main1.cpp > ...

```cpp
#include <iostream>
#include "stack.h"

using namespace std;

int main() {
    cout << "Hello world!" << endl;
    Stack S;

    createStack(S);
    push(S, 3);
    push(S, 4);
    push(S, 8);
    pop(S);
    push(S, 2);
    push(S, 3);
    pop(S);
    push(S, 9);
    printInfo(S);
    cout << "balik stack" << endl;
    balikStack(S);
    printInfo(S);
    return 0;
}
```

*Untitl...

File Edit Format View Help

103112430133
Faris Walid Awwal Aidi

Windows (CRLF) UTF-8

- main.cpp untuk nomor 2

main2.cpp > ...

```cpp
#include <iostream>
#include "stack.h"

using namespace std;

int main() {
    cout << "Halo Semua" << endl;
    Stack S;
    createStack(S);
    pushAscending(S, 3);
    pushAscending(S, 4);
    pushAscending(S, 8);
    pushAscending(S, 2);
    pushAscending(S, 3);
    pushAscending(S, 9);
    printInfo(S);
    cout << "balik stack" << endl;
    balikStack(S);
    printInfo(S);
    return 0;
}
```

*Untitl...

File Edit Format View Help

103112430133
Faris Walid Awwal Aidi

Windows (CRLF) UTF-8

- main.cpp untuk nomor 3

main3.cpp > ...

```cpp
#include <iostream>
#include "stack.h"

using namespace std;

int main() {
    cout << "Halo Semua" << endl;
    Stack S;
    createStack(S);
    getInputStream(S);

    printInfo(S);
    cout << "balik stack" << endl;
    balikStack(S);
    printInfo(S);
    return 0;
}
```

*Untitl...

File Edit Format View Help

103112430133
Faris Walid Awwal Aidi

Windows (CRLF) UTF-8

Screenshots Output 1

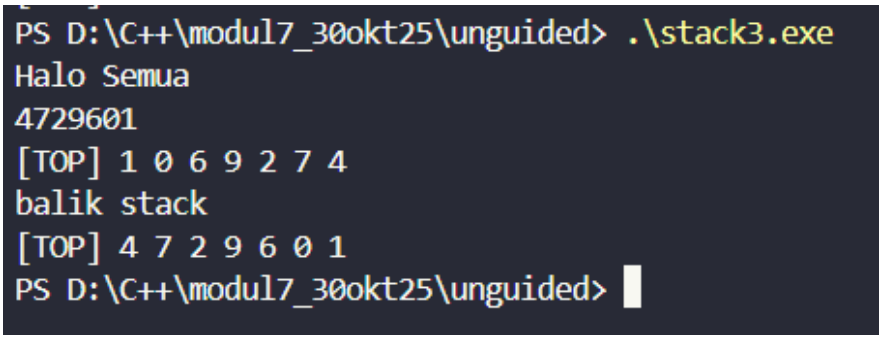
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS D:\C++\modul7_30okt25\unguided> .\stack1.exe
Hello world!
[TOP] 9 2 4 3
balik stack
[TOP] 3 4 2 9
PS D:\C++\modul7_30okt25\unguided>
```

Screenshots Output 2

```
PS D:\C++\modul7_30okt25\unguided> .\stack2.exe
Halo Semua
[TOP] 9 8 4 3 3 2
balik stack
[TOP] 2 3 3 4 8 9
PS D:\C++\modul7_30okt25\unguided>
```

Screenshots Output 3



Deskripsi:

Program ini merupakan implementasi dari struktur data Stack (Tumpukan) dalam bahasa C++ dengan menggunakan representasi tabel, yaitu array berukuran tetap. Program ini mencakup fungsi-fungsi dasar operasional Stack sesuai prinsip LIFO (Last In First Out), seperti createStack untuk inisialisasi tumpukan kosong, push untuk menambahkan elemen ke puncak tumpukan, dan pop untuk mengambil elemen teratas. Selain itu, program ini dilengkapi dengan beberapa fungsi tambahan dari latihan, yaitu printInfo untuk menampilkan semua isi tumpukan, balikStack untuk membalik urutan seluruh elemen di dalam tumpukan, pushAscending untuk memasukkan elemen baru namun tetap menjaga tumpukan terurut menaik, dan getInputStream untuk membaca serangkaian masukan angka dari pengguna hingga tombol 'Enter' ditekan.

D. Kesimpulan

Praktikum Modul 7 ini berhasil mengimplementasikan konsep struktur data Stack (Tumpukan) serta prinsip fundamentalnya, yaitu LIFO (Last In First Out). Pemahaman ini dicapai melalui implementasi dua pendekatan yang berbeda: representasi pointer (menggunakan linked list) dan representasi tabel (menggunakan array). Pada kedua metode, operasi dasar stack yaitu Push untuk menyisipkan elemen di puncak tumpukan dan Pop untuk mengambil elemen dari puncak tumpukan berhasil diterapkan. Perbedaan utama dari kedua pendekatan ini terletak pada manajemen memori dan pengaksesan; implementasi pointer bersifat dinamis, sementara implementasi tabel bersifat statis dengan memanfaatkan pergeseran indeks Top.

Lebih lanjut, praktikum ini juga melatih kemampuan problem solving dengan menerapkan fungsi-fungsi tambahan yang memanipulasi stack, seperti printInfo, balikStack, pushAscending, dan getInputStream. Pembuatan fungsi-fungsi ini membuktikan pemahaman yang lebih dalam mengenai bagaimana struktur LIFO dapat dimanfaatkan untuk operasi yang lebih kompleks, seperti membalik urutan data atau bahkan melakukan penyisipan secara terurut. Latihan ini menunjukkan bahwa dengan memahami batasan akses stack yang hanya pada elemen Top, praktikan dapat membangun logika dan algoritma yang efisien untuk berbagai kebutuhan.

Referensi

Raharjo, Budi. 2025. *Buku Pemrograman C++ Mudah dan Cepat Menjadi Master C.*

Wikipedia contributors. (2024, 8 Mei). C++. Wikipedia, Ensiklopedia Bebas. Diakses pada 2 Oktober 2025, dari https://id.wikipedia.org/wiki/C%2B%2B