

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL VII
STACK**



Disusun Oleh :

NAMA : ABYAN RAHMAN AL FARIZ

NIM : 103112430021

Dosen

FAHRUDIN MUKTI WIBOWO

**PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

Stack adalah sebuah kumpulan data dimana data yang diletakkan di atas data yang lain. Dengan demikian stack adalah struktur data yang menggunakan konsep LIFO (Last In First Out). Dengan demikian, elemen terakhir yang disimpan dalam stack menjadi elemen pertama yang diambil. Dalam proses komputasi, untuk meletakkan sebuah elemen pada bagian atas dari stack, maka dilakukan operasi push. Dan untuk memindahkan dari tempat yang atas tersebut, maka dilakukan operasi pop.

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

Guided 1 (stack.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 bisa pop!" << endl;
        return 0;
    }

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

    delete temp;
    return poppedData;
}

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;
}

```

Screenshots Output:

```

PS D:\TelkomUniversity\Mata Kuliah\Semester 3\Struktur Data\Praktikum> & 'c:\Users\ACER\.vscode\extensions\ms-vscode.cpptools-1.28.3-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-0uhqng1.zfs' '--stdout=Microsoft-MIEngine-Out-sobwt45c.brs' '--stderr=Microsoft-MIEngine-Error-32d5nb5o.5gn' '--pid=Microsoft-MIEngine-Pid-y2uqmxcb.keh' '--dbgExe=C:\Users\ACER\mingw32\bin\gdb.exe' '--interpreter=mi'
Menampilkan isi stack:
TOP -> 30 -> 20 -> 10 -> NULL
Pop: 30
Menampilkan sisa stack:
TOP -> 20 -> 10 -> NULL

```

Deskripsi:

Program di atas merupakan implementasi struktur data Stack (tumpukan) menggunakan Linked List. Stack adalah struktur data dengan prinsip LIFO (Last In, First Out), artinya data yang terakhir masuk akan menjadi data yang pertama keluar. Program ini menggunakan struct Node sebagai elemen stack, di mana setiap node menyimpan sebuah nilai data dan pointer next yang menunjuk ke elemen berikutnya dalam stack. Variabel top digunakan sebagai penanda elemen teratas stack.

Fungsi isEmpty() digunakan untuk memeriksa apakah stack masih kosong. Fungsi push() menambahkan elemen baru ke bagian atas stack dengan membuat node baru dan menautkannya ke node sebelumnya. Fungsi pop() digunakan untuk menghapus elemen teratas stack, sekaligus mengembalikan nilai yang dihapus. Jika stack kosong, fungsi akan menampilkan pesan peringatan. Selanjutnya, fungsi show() menampilkan semua elemen stack dari posisi TOP hingga elemen paling bawah, yang diakhiri dengan NULL.

Pada fungsi main(), program membuat stack kosong, kemudian menambahkan beberapa data (10, 20, 30) menggunakan push(). Setelah itu, isi stack ditampilkan, satu elemen di-pop, dan isi stack ditampilkan kembali. Dengan demikian, program ini menunjukkan cara kerja stack secara dinamis menggunakan linked list, lengkap dengan operasi dasar push, pop, dan display.

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

Unguided 1 (stack.cpp)

```
#include "Stack.h"

void createStack(Stack &S) {
    S.top = -1;
}

void push(Stack &S, infotype x) {
    if (S.top < 19) {
        S.top++;
        S.info[S.top] = x;
    } else {
        cout << "Stack penuh!" << endl;
    }
}

infotype pop(Stack &S) {
    if (S.top >= 0) {
        infotype x = S.info[S.top];
        S.top--;
        return x;
    } else {
        cout << "Stack kosong!" << endl;
        return -1;
    }
}

void printInfo(Stack S) {
    cout << "[TOP] ";
    for (int i = S.top; i >= 0; i--) {
        cout << S.info[i] << " ";
    }
    cout << endl;
}

void balikStack(Stack &S) {
    Stack temp;
```

```

        createStack(temp);

        while (S.top >= 0) {
            push(temp, pop(S));
        }

        S = temp;
    }
}

```

Unguided 1 (stack.h)

```

#ifndef STACK_H_INCLUDED
#define STACK_H_INCLUDED

#include <iostream>
using namespace std;

typedef int infotype;

struct Stack {
    infotype info[20];
    int top;
};

void createStack(Stack &S);
void push(Stack &S, infotype x);
infotype pop(Stack &S);
void printInfo(Stack S);
void balikStack(Stack &S);

#endif

```

Unguided 1 (main.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;
}

```

Screenshots Output:

```

PS D:\TelkomUniversity\Mata Kuliah\Semester 3\Struktur Data\Praktikum\Modul 7 - Stack\Unguided> & 'c:\Users\ACER\.vscode\extensions\ms-vscode.cpptools-1.28.3-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-fdw4af0t.v52' '--stdout=Microsoft-MIEngine-Out-x3ne2npm.3bb' '--stderr=Microsoft-MIEngine-Error-ecvil4po.beo' '--pid=Microsoft-MIEngine-Pid-m30ezi1u.pg1' '--dbgExe=C:\Users\ACER\mingw32\bin\gdb.exe' '--interpreter=mi'
Hello world!
[TOP] 9 2 4 3
balik stack
[TOP] 3 4 2 9

```

Deskripsi:

Program ini menerapkan ADT Stack menggunakan Array sebagai struktur penyimpanan utama. Stack bekerja dengan prinsip LIFO (Last In, First Out), di mana data yang terakhir dimasukkan akan menjadi yang pertama dikeluarkan. Pada implementasi ini, tipe data stack didefinisikan di dalam file Stack.h, yang berisi array info[20] sebagai tempat penyimpanan data serta variabel top yang digunakan untuk menandai posisi elemen paling atas di dalam stack.

Fungsi createStack() digunakan untuk menginisialisasi stack dengan memberi nilai awal top = -1, menandakan bahwa stack masih kosong. Fungsi push() bertugas menambah elemen baru ke posisi paling atas apabila kapasitas stack belum penuh. Sebaliknya, fungsi pop() mengambil dan menghapus elemen paling atas, sekaligus mengembalikan nilainya. Jika stack kosong, fungsi akan memberikan pesan peringatan.

Adapun `printInfo()` digunakan untuk menampilkan isi stack mulai dari elemen teratas hingga terbawah, sehingga memudahkan pengguna melihat kondisi stack saat ini.

Program ini juga menyertakan fungsi `balikStack()`, yaitu proses untuk membalik urutan elemen di dalam stack. Mekanismenya adalah dengan memindahkan seluruh isi stack utama ke stack sementara menggunakan operasi `pop` dan `push`, kemudian menggantikan isi stack asli dengan stack yang sudah dibalik.

Pada file `main.cpp`, program melakukan serangkaian operasi seperti `push`, `pop`, menampilkan isi stack, hingga membalik isi stack. Hasil akhirnya menunjukkan bagaimana struktur data stack bekerja dalam skenario nyata, mulai dari penyimpanan, penghapusan elemen, hingga pembalikan urutan data menggunakan array sebagai dasar implementasinya.

D. Kesimpulan

Stack merupakan salah satu struktur data fundamental yang bekerja dengan prinsip LIFO (Last In, First Out), di mana elemen yang terakhir dimasukkan akan menjadi elemen pertama yang diambil. Konsep ini membuat stack sangat efisien untuk operasi yang membutuhkan pengelolaan data secara berurutan dan terbalik, seperti proses `undo/redo`, evaluasi ekspresi, hingga pemanggilan fungsi (`call stack`) pada program.

Dalam implementasinya, stack dapat dibangun menggunakan array maupun linked list. Pada materi ini, stack diimplementasikan menggunakan array, dengan variabel `top` sebagai penanda posisi elemen teratas. Operasi dasar dalam stack meliputi `push` (menambah elemen), `pop` (menghapus elemen teratas), serta `print` untuk menampilkan isi stack. Selain itu, fungsi tambahan seperti `balikStack` menunjukkan bagaimana elemen di dalam stack dapat dibalik menggunakan operasi dasar `push` dan `pop`.

E. Referensi

<https://www.nblognlife.com/2014/04/stack-pada-c.html>

<https://www.trivusi.web.id/2022/07/struktur-data-stack.html>

<https://daismabali.medium.com/mengenal-konsep-stack-dalam-struktur-data-beserta-ilustrasi-dan-contoh-pemrogramannya-e63e444886e9>