

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL VII
STACK**



Disusun Oleh :

NAMA : Damanik, Yohanes Geovan Ondova
NIM : 103112400022

Dosen

FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

- *Stack* adalah salah satu bentuk struktur data yang prinsip.
- Elemen yang pertama kali bisa diambil adalah elemen yang paling atas.
- Prinsip operasi pada *stack* disebut LIFO (*Last In First Out*).
- Komponen utama dalam *stack* yang berfungsi untuk mengakses data disebut “Top”

Operasi Utama dalam Stack (Representasi Pointer):

- Push: Operasi menyisipkan elemen pada tumpukan data, mirip dengan fungsi *insert first* pada *list* biasa.
- Pop: Operasi pengambilan data dalam *list*, mirip dengan *delete first*. Elemen yang diakses adalah elemen paling atas.

Primitif dalam Stack

Primitif *stack* lebih sedikit daripada *list* lainnya karena hanya beroperasi pada elemen paling atas. Contoh primitif: *createStack()*, *isEmpty()*, *alokasi()*, *dealokasi()*, dan fungsi-fungsi pencarian.

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;
    }
}

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;
}

```

Output

```
PS C:\Users\Lenovo\Documents\STURKTUR DATA MODUL 7> cd "c:\Users\Lenovo\Documents\STU
) { g++ stack.cpp -o stack } ; if ($?) { .\stack }
Menampilkan isi stack:
TOP -> 30 -> 20 -> 10 -> Pop: 30
Menampilkan sisa stack:
TOP -> 20 -> 10 ->
PS C:\Users\Lenovo\Documents\STURKTUR DATA MODUL 7>
```

Deskripsi:

Program tersebut merupakan implementasi struktur data **stack (tumpukan)** menggunakan linked list dalam C++, di mana setiap elemen stack direpresentasikan sebagai node dengan dua komponen yaitu data (nilai integer) dan pointer ke node berikutnya. Program memiliki fungsi-fungsi dasar stack seperti isEmpty untuk memeriksa kekosongan stack, push untuk menambahkan elemen di atas tumpukan (TOP), pop untuk menghapus dan mengembalikan elemen teratas, serta show untuk menampilkan seluruh isi stack. Pada fungsi main, program mendemonstrasikan operasi dengan menambahkan tiga nilai (10, 20, 30) ke stack, menampilkan isinya, lalu melakukan pop untuk menghapus elemen teratas (30) dan menampilkan kembali sisa stack (20 → 10)

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;
}

bool isEmpty(Stack S) {
    return (S.top == -1);
}

bool isFull(Stack S) {
    return (S.top == nMax - 1);
}

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

```

}

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

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

void balikStack(Stack &S) {
    if (!isEmpty(S)) {
        int temp[nMax];
        int count = 0;
        while (!isEmpty(S)) {
            temp[count] = pop(S);
            count++;
        }

        for (int i = 0; i < count; i++) {
            push(S, temp[i]);
        }
    }
}

void pushAscending(Stack &S, infotype x) {
    if (isEmpty(S) || x >= S.info[S.top]) {
        push(S, x);
    } else {
        infotype temp = pop(S);
        pushAscending(S, x);
        push(S, temp);
    }
}

```

```

void getInputStream(Stack &S) {
    char input;
    while (cin.get(input) && input != '\n') {
        if (input >= '0' && input <= '9') {
            int val = input - '0';
            push(S, val);
        }
    }
}

```

stack.h

```

#ifndef STACK_H
#define STACK_H

#include <iostream>
using namespace std;

#define nMax 20
typedef int infotype;

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

void createStack(Stack &S);
bool isEmpty(Stack S);
bool isFull(Stack S);
void push(Stack &S, infotype x);
infotype pop(Stack &S);
void printInfo(Stack S);
void balikStack(Stack &S);
void pushAscending(Stack &S, infotype x);
void getInputStream(Stack &S);

#endif

```

Main.cpp

```

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

using namespace std;

int main() {
    Stack S;
    createStack(S);
}

```

```

    cout << "Hello World!" << endl;
    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);

    while(!isEmpty(S)) { pop(S); }

    cout << "\n" << endl;
    cout << "Masukkan angka: ";

    getInputStream(S);
    printInfo(S);
    cout << "balik stack" << endl;
    balikStack(S);
    printInfo(S);

    return 0;
}

```

Output

```

PS C:\Users\Lenovo\Documents\STURKTUR DATA MODUL 7> cd "c:\Users\Lenovo\Documents\STURKTU
) { g++ main.cpp -o main } ; if ($?) { .\main }
Hello World!
[TOP] 9 8 4 3 3 2
balik stack
[TOP] 2 3 3 4 8 9

Masukkan angka: 4729601
[TOP] 1 0 6 9 2 7 4
balik stack
[TOP] 4 7 2 9 6 0 1
PS C:\Users\Lenovo\Documents\STURKTUR DATA MODUL 7> 

```

Deskripsi:

Program ini merupakan implementasi lengkap struktur data **stack** menggunakan array statis yang terbagi dalam tiga file: `stack.h` untuk deklarasi struktur dan fungsi, `stack.cpp` untuk implementasi fungsi-fungsinya, dan `main.cpp` sebagai program utama yang mendemonstrasikan berbagai operasi stack. Program mendefinisikan stack dengan kapasitas maksimum 20 elemen dan menyediakan fungsi dasar

seperti push, pop, isEmpty, dan isFull, serta fungsi tambahan seperti balikStack untuk membalik urutan elemen, pushAscending untuk memasukkan data secara terurut ascending, dan getInputStream untuk membaca input angka dari pengguna. Pada main.cpp, program menunjukkan penggunaan fungsi-fungsi tersebut dengan mengisi stack secara terurut, membalik isinya, menerima input manual, dan menampilkan hasil setiap operasi menggunakan printInfo.

D. Kesimpulan

Laporan praktikum modul stack ini berhasil mengimplementasikan dua jenis struktur data stack menggunakan representasi yang berbeda, yaitu linked list pada bagian guided dan array statis pada bagian unguided. Program guided mendemonstrasikan operasi dasar stack (push, pop, isEmpty, show) dengan prinsip LIFO, sementara program unguided menambahkan fungsi lanjutan seperti pembalikan stack, penyisipan ascending, dan penerimaan input stream. Kedua implementasi ini secara komprehensif menunjukkan karakteristik fundamental stack beserta variasinya, sekaligus membuktikan fleksibilitas struktur data stack dalam menyelesaikan berbagai masalah pemrosesan data dengan efisiensi yang baik..

E. Referensi

<https://www.geeksforgeeks.org/cpp/stack-in-cpp-stl/>

<https://www.geeksforgeeks.org/cpp/stack-push-and-pop-in-c-stl/>

https://www.w3schools.com/cpp/cpp_functions.asp