

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

MODUL 7



Disusun Oleh :
NAMA : IBTIDA ZADA UTOMO
NIM : 103112430037

Dosen
FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

Struktur data merupakan cara untuk mengatur dan menyimpan data agar dapat digunakan secara efisien oleh program komputer. Salah satu struktur data yang paling umum digunakan adalah stack (tumpukan). Stack bekerja dengan prinsip LIFO (Last In, First Out), artinya elemen yang terakhir dimasukkan akan menjadi elemen pertama yang dikeluarkan. Stack banyak digunakan dalam proses seperti pemanggilan fungsi rekursif, pembalikan urutan data, serta pengelolaan memori.

Pada implementasinya, stack dapat dibuat menggunakan array maupun linked list. Implementasi berbasis array memiliki ukuran tetap yang dibatasi oleh kapasitas maksimum (misalnya MAX = 20), sedangkan implementasi linked list bersifat dinamis karena dapat menambah node baru selama memori masih tersedia.

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

Guided 1

```
#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;
    cout << data << " pushed to stack\n";
}
```

```
int pop(Node *&top)
{
    if (isEmpty(top))
    {
        cout << "Stack kosong, tidak bisa pop\n";
        return 0;
    }
    Node *temp = top;
    int poppedData = top->data;
    top = top->next;

    delete temp;
    return poppedData;
}

void show(Node *top)
{
    if (isEmpty(top))
    {
        cout << "Stack kosong\n";
        return;
    }
    cout << "Elemen teratas adalah " << top->data << endl;
    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 isi stack setelah pop: " << endl;
show(stack);

return 0;
}

```

Screenshots Output

```

o stack } ; if ($?) { .\stack }
10 pushed to stack
20 pushed to stack
30 pushed to stack
menampilkan isi stack :
Elemen teratas adalah 30
30 -> 20 -> 10 -> NULL
Pop : 30
menampilkan isi stack setelah pop:
Elemen teratas adalah 20
20 -> 10 -> NULL
PS D:\struktur data\MODUL 7\GUIDED>

```

Deskripsi:

program diatas adalah program untuk implementasi stack di struktur data menggunakan linked list di bahasa c++ berbeda dengan stack di array program ini menyimpan element secara dinamis di memori menggunakan node yang saling terhubung

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

Unguided 1

stack.h

```

#ifndef STACK_H
#define STACK_H

```

```

const int MAX = 20;

typedef int infotype;

struct Stack {
    infotype info[MAX];
    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

```

stack.cpp

```

#include <iostream>
#include "stack.h"
using namespace std;

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

void push(Stack &S, infotype x) {
    if (S.top < MAX - 1) {
        S.top++;
        S.info[S.top] = x;
    } else {
        cout << "Stack overflow!" << endl;
    }
}

infotype pop(Stack &S) {
    if (S.top >= 0) {
        infotype x = S.info[S.top];
        S.top--;
        return x;
    } else {

```

```

        cout << "Stack underflow!" << endl;
        return -1; // nilai error
    }

}

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;
}

```

main.cpp

```

#include <iostream>
#include "stack.h"
#include "stack.cpp"
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:\struktur data\MODUL 7\UNGUIDED> cd "d:\struktur data\MODUL 7\UNGUIDED\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
Hello world!
[TOP] 9 2 4 3
balik stack
[TOP] 3 4 2 9
PS D:\struktur data\MODUL 7\UNGUIDED> █

```

Deskripsi: program diatas adalah program untuk mendefinisikan struktur data dan deklarasi fungsi yang digunakan dalam program stack di dalamnya juga terdapat fungsi dasar pada stack yaitu createstack, push, pop, printinfo, dan balikstack

Unguided 2

stack2.h

```

#ifndef STACK_H
#define STACK_H

const int MAX = 20;
typedef int infotype;

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

void createStack(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);

```

```
#endif
```

stack2.cpp

```
#include <iostream>
#include "stack2.h"
using namespace std;

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

void push(Stack &S, infotype x) {
    if (S.top < MAX - 1) {
        S.top++;
        S.info[S.top] = x;
    } else {
        cout << "Stack overflow!" << endl;
    }
}

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

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;
    }

void pushAscending(Stack &S, infotype x) {
    if (S.top >= MAX - 1) {
        cout << "Stack overflow!" << endl;
        return;
    }

    Stack temp;
    createStack(temp);

    while (S.top >= 0 && S.info[S.top] > x) {
        push(temp, pop(S));
    }

    push(S, x);

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

```

main2.cpp

```

#include <iostream>
#include "stack2.h"
#include "stack2.cpp"
using namespace std;

int main() {
    cout << "Hello world!" << 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;
}

```

Screenshots Output

```

PS D:\struktur data\MODUL 7\UNGUIDED> cd "d:\struktur data\MODUL 7\UNGUIDED\" ; if ($?) { g++ main2.cpp -o main2 } ; if ($?) { ./main2 }
Hello world!
[TOP] 9 8 4 3 3 2
balik stack
[TOP] 2 3 3 4 8 9
PS D:\struktur data\MODUL 7\UNGUIDED> []

```

Deskripsi: program diatas adalah program untuk mengimplementasikan adt atau abstract data type yang menggunakan array di bahasa c++, struktur tersebut dibagi menjadi 3 file yaitu stack2.h, stack2.cpp, dan main2.cpp nantinya program akan menampilkan hello eord dan angka random lalu di balik stack jadi angkanya akan terurut terbalik

Unguided 3

stack3.h

```

#ifndef STACK_H
#define STACK_H

#include <iostream>
using namespace std;

```

```

const int MAX = 20;
typedef int infotype;

struct Stack {
    infotype info[MAX];
    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 getInputStream(Stack &S);

#endif

```

stack3.cpp

```

#include "stack3.h"
#include <cctype>
#include <iostream>

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

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

bool isFull(Stack S) {
    return (S.top == MAX - 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) {
    if (!isEmpty(S)) {
        int 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] (kosong)" << endl;
    } else {
        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 (!isEmpty(S)) {
        push(temp, pop(S));
    }
    S = temp;
}

void getInputStream(Stack &S) {
    cout << "Masukkan angka (akhiri dengan ENTER): ";
    char c;
    while (true) {
        c = cin.get();
        if (c == '\n') break;
        if (isdigit(c)) {

```

```

        int num = c - '0';
        push(S, num);
    }
}

```

main3.cpp

```

#include <iostream>
#include "stack3.h"
#include "stack3.cpp"
using namespace std;

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

    Stack S;
    createStack(S);

    getInputStream(S);
    printInfo(S);

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

    return 0;
}

```

Screenshots Output

```

PS D:\struktur data\MODUL 7\UNGUIDED> cd "d:\struktur data\MODUL 7\UNGUIDED\" ; if ($?) { g++ main3.cpp -o main3 } ; if ($?) { ./main3 }
Hello world!
Masukkan angka (akhiri dengan ENTER): 4729601
[TOP] 1 0 6 9 2 7 4
balik stack
[TOP] 4 7 2 9 6 0 1
PS D:\struktur data\MODUL 7\UNGUIDED>

```

Deskripsi: program diatas adalah program untuk implementasi array di bahasa c++ fungsi utama programnya yaitu untuk membaca input angka dari pengguna, lalu menyimpannya ke dalam stack, dan menampilkan isi stack, kemudian membalik urutan stack dan menampilkan hasilnya lagi, program diatas nanti meminta kita meng inputkan angka angka dan program akan menampilkan terurut dan ada spasinya lalu program akan membalikan angka angka tersebut dari paling belakang ke paling depan

C. Kesimpulan

Berdasarkan hasil percobaan dan implementasi program, dapat disimpulkan bahwa stack merupakan struktur data yang sangat penting karena memudahkan pengelolaan data dengan konsep LIFO. Melalui penggunaan ADT Stack pada C++, program dapat melakukan berbagai operasi seperti memasukkan data (push), menghapus data (pop), menampilkan isi stack (printInfo), membalik urutan elemen (balikStack), serta membaca input dinamis dari pengguna (getInputStream).

Implementasi menggunakan array menunjukkan cara kerja stack dengan kapasitas terbatas, sedangkan implementasi menggunakan linked list menunjukkan cara penyimpanan data yang lebih fleksibel tanpa batas ukuran tetap. Dengan pendekatan modular yang membagi kode menjadi tiga file utama (stack.h, stack.cpp, dan main.cpp), program menjadi lebih terstruktur, mudah dibaca, dan efisien untuk dikembangkan.

Secara keseluruhan, program ini berhasil menggambarkan prinsip kerja stack, baik dalam pengelolaan data secara statis maupun dinamis, serta memperlihatkan bagaimana struktur data dapat diimplementasikan secara praktis untuk memecahkan berbagai permasalahan pemrograman.

D. Referensi

- <https://www.belajarcpp.com/referensi/>
- <https://www.w3schools.com/cpp/default.asp>
- <https://www.petanikode.com/cpp-untuk-pemula/>