

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

MODUL 7



Disusun Oleh :
NAMA : Herdian Abdillah Purnomo
NIM : 103112430048

Dosen
FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

Struktur data adalah cara mengatur dan menyimpan data agar program dapat mengakses dan mengolahnya dengan efisien. Salah satu struktur data yang sering digunakan adalah stack (tumpukan). Stack bekerja dengan konsep LIFO (Last In, First Out), yaitu data yang dimasukkan terakhir akan keluar terlebih dahulu. Struktur ini banyak digunakan dalam proses seperti pemanggilan fungsi rekursif, membalik data, dan pengelolaan memori.

Stack dapat dibuat menggunakan array atau linked list. Jika menggunakan array, ukuran stack sudah ditentukan sejak awal (misalnya MAX = 20). Sedangkan dengan linked list, ukuran stack dapat bertambah secara dinamis 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

```

PS C:\Users\Lenovo\Documents\PRAKTIKUM STRUKDAT\FILE\MODUL 7\UNGUIDED\SOAL 3> cd "c:\Users\Lenovo\Documents\PRAKTIKUM STRUKDAT\FILE\MODUL 7\GUIDED\" ; if ($?) {
  g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile ; if ($?) { ./tempCodeRunnerFile }

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 C:\Users\Lenovo\Documents\PRAKTIKUM STRUKDAT\FILE\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.

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

Unguided 1

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;
}

```

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;
}

```

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
```

Screenshots Output

```
PS C:\Users\Lenovo\Documents\PRAKTIKUM STRUKDAT\FILE\MODUL 7\GUIDED> cd "c:\Users\Lenovo\Documents\PRAKTIKUM STRUKDAT\FILE\MODUL 7\UNGUIDED\SOAL 1"
Hello world!
[TOP] 9 2 4 3
balik stack
[TOP] 3 4 2 9
PS C:\Users\Lenovo\Documents\PRAKTIKUM STRUKDAT\FILE\MODUL 7\UNGUIDED\SOAL 1>
```

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

main.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;
}
```

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));
}
}

```

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

```

Screenshots Output

```
PS C:\Users\Lenovo\Documents\PRAKTIKUM STRUKDAT\FILE\MODUL 7\UNGUIDED\SOAL 1> cd "c:\Users\Lenovo\Documents\PRAKTIKUM STRUKDAT\FILE\MODUL 7\UNGUIDED\SOAL 2"
Hello world!
[TOP] 9 8 4 3 3 2
balik stack
[TOP] 2 3 3 4 8 9
PS C:\Users\Lenovo\Documents\PRAKTIKUM STRUKDAT\FILE\MODUL 7\UNGUIDED\SOAL 2>
```

Deskripsi:

Program diatas adalah program untukmengimplementasikan 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

main.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;
}
```

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);
    }
}

```

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

```

Screenshots Output

```
PS C:\Users\Lenovo\Documents\PRAKTIKUM STRUKDAT\FILE\MODUL 7\UNGUIDED\SOAL 3> cd "c:\Users\Lenovo\Documents\PRAKTIKUM STRUKDAT\FILE\MODUL 7\UNGUIDED\SOAL 3\" ; if ($?) { g++ main.cpp -o main } ; if ($?) { .\main }
Hello world!
Masukkan angka (akhiri dengan ENTER): 040306
[TOP] 6 0 3 0 4 0
balik stack
[TOP] 0 4 0 3 0 6
PS C:\Users\Lenovo\Documents\PRAKTIKUM STRUKDAT\FILE\MODUL 7\UNGUIDED\SOAL 3>
```

Deskripsi:

Program diatas adalah program untuk implementasi array di bahasa c++ fungsi utama programnya yaitu untuk membaca input angka dari pengguna, lalu menyimpanya 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 spasiannya lalu program akan membalikan angka angka tersebut dari paling belakang ke paling depan.

E. Kesimpulan

Berdasarkan percobaan dan program yang dibuat, dapat disimpulkan bahwa stack adalah struktur data penting yang bekerja dengan prinsip LIFO (Last In First Out). Dengan ADT Stack pada C++, program dapat melakukan operasi seperti menambah data (push), menghapus data (pop), menampilkan isi stack, membalik isi stack, dan menerima input dari pengguna.

Penggunaan array menunjukkan stack dengan kapasitas terbatas, sedangkan linked list memberi fleksibilitas tanpa batas ukuran tertentu. Pembagian kode menjadi tiga file (stack.h, stack.cpp, dan main.cpp) membuat program lebih terstruktur dan mudah dipahami.

Secara keseluruhan, program ini berhasil menunjukkan cara kerja stack baik secara statis maupun dinamis, serta penerapannya dalam pemrograman untuk menyelesaikan berbagai masalah.

F. Referensi

Weiss, M. A. (2014). *Data Structures and Algorithm Analysis in C++*. Pearson.

Malik, D. S. (2010). *Data Structures Using C++*. Course Technology.

Sedgewick, R., & Wayne, K. (2011). *Algorithms*. Addison-Wesley.

Stroustrup, B. (2013). *The C++ Programming Language*. Addison-Wesley.

GeeksforGeeks. (n.d.). *Stack in C++*. Diakses dari <https://www.geeksforgeeks.org>

Tutorialspoint. (n.d.). *C++ Stack*. Diakses dari <https://www.tutorialspoint.com>