

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL 7
STACK**



Disusun Oleh :

NAMA : Maulana Ananta Piliang
NIM : 103112400156

Dosen
FAHRUDIN MUKTI WIBOWO

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

A. Dasar Teori

C++ adalah pengembangan dari bahasa C yang dibuat oleh Bjarne Stroustrup sekitar tahun 1980-an. C++ disebut bahasa multi-paradigma, artinya bisa dipakai dengan gaya prosedural (pakai fungsi biasa), berorientasi objek (pakai class dan object), atau bahkan gabungan keduanya. C++ punya dasar-dasar seperti variabel, operator percabangan (if, switch), perulangan (for, while), dan bisa memakai class untuk membuat objek.

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

Guided1

h

```
#ifndef STACK_H
#define STACK_H

#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
```

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;

    }

}

infotype pop(Stack &S) {

    infotype x = -1;

    if (S.top >= 0) {

        x = S.info[S.top];
        S.top--;
    }

    return x;
}
```

```
}
```



```
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 T;
```



```
    createStack(T);
```



```
    while (S.top >= 0) {
```



```
        push(T, pop(S));
```



```
    }
```



```
    S = T;
```



```
}
```

Main

```
#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, 2);
    push(S, 9);

    pop(S);
    pop(S);
    pop(S);

    push(S, 5);
    push(S, 1);

    printInfo(S);

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

```
    printInfo(S);

    return 0;
}
```

Screenshots Output

```
>> C:\STRUKTUR DATA\laprak modul 7>
Hello world!
[TOP] 1 5 3
balik stack
[TOP] 3 5 1
PS C:\STRUKTUR DATA\laprak modul 7> █
```

Deskripsi:

Program ini mengimplementasikan struktur data Stack menggunakan array dengan kapasitas 20 elemen. Operasi yang disediakan meliputi push untuk menambahkan data, pop untuk menghapus data teratas, printInfo untuk menampilkan isi stack, dan balikStack untuk membalik urutan elemen dalam stack. Pada fungsi main, beberapa nilai dimasukkan ke dalam stack, kemudian sebagian dihapus, dan setelah itu stack ditampilkan sebelum dan sesudah proses pembalikan. Program ini menunjukkan cara kerja stack dengan konsep LIFO (Last In, First Out).

Guided 2

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

const int MAX = 100;

struct Stack {
    int data[MAX];
    int top = -1;
};

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

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

void push(Stack &S, int x) {
S.data[++S.top] = x; }

int pop(Stack &S) { return
S.data[S.top--]; }

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

// push ascending

void pushAscending(Stack &S, int
x) {
    Stack temp;
```

```
// pindah elemen yang lebih kecil  
dari x
```

```
while (!isEmpty(S) &&  
S.data[S.top] < x) {  
    push(temp, pop(S));  
}
```

```
push(S, x); // masukkan x
```

```
// kembalikan elemen dari temp
```

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

```
// membalik stack
```

```
void balik(Stack &S) {  
    Stack temp;  
    while (!isEmpty(S)) push(temp,  
pop(S));  
    S = temp;  
}
```

```
int main() {  
    cout << "Hello world!" << endl;  
// <--- sudah ditambahkan
```

```
Stack S;
```

```
pushAscending(S, 3);  
pushAscending(S, 4);
```

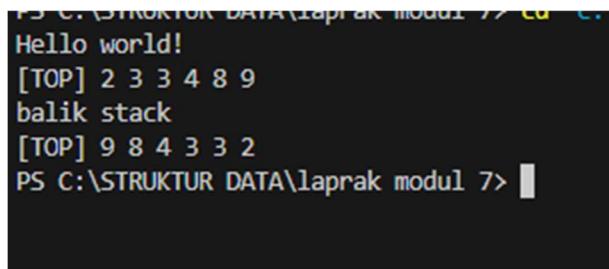
```
pushAscending(S, 8);
pushAscending(S, 2);
pushAscending(S, 3);
pushAscending(S, 9);

print(S);

cout << "balik stack" << endl;
balik(S);
print(S);

return 0;
}
```

Screenshots Output



```
PS C:\STRUKTUR DATA\laprak modul 7> cd ..\..
Hello world!
[TOP] 2 3 3 4 8 9
balik stack
[TOP] 9 8 4 3 3 2
PS C:\STRUKTUR DATA\laprak modul 7>
```

Deskripsi:

Program ini menggunakan stack untuk menyimpan data dan menyediakan fungsi pushAscending agar data selalu masuk dalam urutan menaik. Program juga bisa membalik isi stack dan menampilkan hasil sebelum dan sesudah dibalik.

Guided 3

```
#include <iostream>

using namespace std;

const int MAX = 100;

struct Stack {

    char data[MAX];

    int top = -1;

};

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

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

void push(Stack &S, char x) {

    if (!isFull(S))

        S.data[++S.top] = x;

}

char pop(Stack &S) {
```

```
    return S.data[S.top--];

}

void printInfo(Stack S) {

    cout << "[TOP] ";

    for (int i = S.top; i >= 0; i--)
    {

        cout << S.data[i] << " ";

    }

    cout << endl;

}

// sesuai soal, memakai
cin.get()

void    getInputStream(Stack
&S) {

    cout << "Masukkan input: ";

    char   c   =   cin.get(); // membaca karakter pertama

    if (c == '\n') // kalau user langsung enter
```

```
c = cin.get(); // baca  
ulang
```

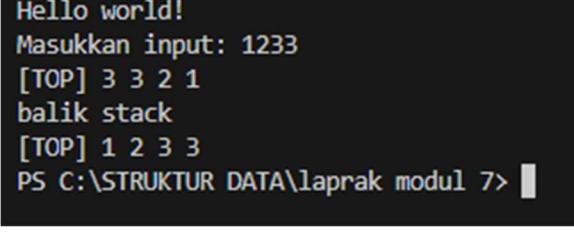
```
while (c != '\n') {  
  
    push(S, c);  
  
    c = cin.get(); // baca  
    karakter berikutnya  
  
}  
  
}
```

```
void balikStack(Stack &S) {  
  
    Stack temp;  
  
    while (!isEmpty(S)) {  
  
        push(temp, pop(S));  
  
    }  
  
    S = temp;  
  
}
```

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

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

Screenshots Output



```
Hello world!  
Masukkan input: 1233  
[TOP] 3 3 2 1  
balik stack  
[TOP] 1 2 3 3  
PS C:\STRUKTUR DATA\laprak modul 7>
```

Deskripsi:

Program membaca input karakter per karakter, menaruhnya ke dalam stack, menampilkan isi stack, lalu membalik urutannya dan menampilkannya kembali.

D. Kesimpulan

Pada praktikum modul 7 ini dapat disimpulkan bahwa struktur data Stack merupakan konsep penting dalam pemrograman yang bekerja dengan prinsip LIFO (Last In, First Out). Melalui tiga program yang telah dibuat, mahasiswa memahami berbagai operasi dasar stack, seperti push, pop, menampilkan isi stack, serta membalik urutan elemen menggunakan stack bantu. Selain itu, praktikum ini memperkenalkan variasi operasi seperti pushAscending yang menjaga agar data masuk dalam urutan tertentu, dan getInputStream yang membaca input karakter demi karakter menggunakan cin.get(). Seluruh latihan ini membantu memperkuat pemahaman mengenai cara kerja stack berbasis array, bagaimana mengelola data secara bertumpuk, serta bagaimana memanfaatkan stack untuk memproses data secara efisien dalam berbagai kasus.

E. Referensi

- Modul Praktikum Struktur Data – Telkom University Purwokerto
 - Goodrich, M., Tamassia, R. (2011). *Data Structures and Algorithms in C++*. Wiley.
 - cppreference.com – C++ Documentation
 - GeeksforGeeks. *Stack Data Structure in C++*.
-