

**LAPORAN PRAKTIKUM STRUKTUR
DATA**

**MODUL VI I
STACK**



Disusun Oleh :
NAMA : IVAN RAMADHAN
NIM : 103112400186

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 1 stack.h

```
#ifndef STACK_H_INCLUDED
#define STACK_H_INCLUDED
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
```

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 < 19) {
        S.top++;
        S.info[S.top] = x;
    }
}
infotype pop(Stack &S) {
    if (S.top >= 0) {
        infotype x = S.info[S.top];
        S.top--;
        return x;
    }
    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 T;
    createStack(T);
    while (S.top >= 0) {
        push(T, pop(S));
    }
    S = T;
}

```

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:

1.

```
    >>> ./stack
>>
Hello world!
[TOP] 9 2 4 3
balik stack
[TOP] 3 4 2 9
PS C:\xampp\htdocs\Modul Alpro\laprakstd>
```

Guided 2_stack.h

```
#ifndef STACK_H_INCLUDED
#define STACK_H_INCLUDED

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);
void pushAscending(Stack &S,
infotype x);

#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 < 19) {
        S.top++;
        S.info[S.top] = x;
    }
}

infotype pop(Stack &S) {
    if (S.top >= 0) {
        infotype x = S.info[S.top];
        S.top--;
        return x;
    }
    return -1;
}

void printInfo(Stack S) {
    cout << "[TOP] ";
    for (int i = S.top; i >= 0; i--) {
        if (S.info[i] >= 48 && S.info[i] <= 57) {
            cout << (char) S.info[i] << " ";
        } else {
            cout << S.info[i] << " ";
        }
    }
    cout << endl;
}

void balikStack(Stack &S) {
    Stack T;
    createStack(T);

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

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

```

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));
}
}

```

Main.cpp

```

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

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:

1.

```
    Hello world!
[TOP] 9 8 4 3 3 2
balik stack
[TOP] 2 3 3 4 8 9
PS C:\xampp\htdocs\Modul Alpro\laprakstd>
```

Guided 3 stack.h

```
#ifndef STACK_H_INCLUDED
#define STACK_H_INCLUDED

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);
void pushAscending(Stack &S,
infotype x);
void getInputStream(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 < 19) {
        S.top++;
        S.info[S.top] = x;
    }
}

infotype pop(Stack &S) {
    if (S.top >= 0) {
        infotype x = S.info[S.top];
        S.top--;
        return x;
    }
    return -1;
}

void printInfo(Stack S) {
    cout << "[TOP] ";
    for (int i = S.top; i >= 0; i--) {
        if (S.info[i] >= 48 && S.info[i] <= 57) {
            cout << (char) S.info[i] << " ";
        } else {
            cout << S.info[i] << " ";
        }
    }
    cout << endl;
}

void balikStack(Stack &S) {
    Stack T;
    createStack(T);

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

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

```

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));
}
}

void getInputStream(Stack &S) {
    char c;
    while (true) {
        c = cin.get();
        if (c == '\n') {
            break;
        }
        push(S, (int) c);
    }
}

```

Main.cpp

```

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

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:

1.

```
Hello world!
8765432
[TOP] 2 3 4 5 6 7 8
balik stack
[TOP] 8 7 6 5 4 3 2
PS C:\xampp\htdocs\Modul Alpro\laprakstd>
```

Deskripsi:

Program ini mengimplementasikan struktur data Stack (tumpukan) menggunakan array statis dengan kapasitas maksimum 20 elemen. Stack bekerja dengan prinsip LIFO (Last In First Out), di mana elemen terakhir yang dimasukkan akan menjadi elemen pertama yang dikeluarkan..

C. Kesimpulan

Program ini berhasil menunjukkan penerapan **struktur data Stack** menggunakan **array statis** dalam bahasa pemrograman C++. Dengan prinsip kerja **LIFO (Last In First Out)**, program mampu melakukan operasi dasar stack seperti inisialisasi, penambahan data (*push*), penghapusan data (*pop*), penampilan isi stack, serta pembalikan urutan elemen stack. Implementasi ini membantu memahami cara pengelolaan data secara terstruktur dan efisien, serta memperlihatkan bagaimana stack dapat digunakan untuk memproses data secara berurutan sesuai dengan konsep tumpukan..

D. Referensi

- W3Resource. (2020). *C++ String Exercises: Convert digit/number to words*.
- GeeksforGeeks. (2020). *Loops in C++ (for, while, do-while)*.