

# **LAPORAN PRAKTIKUM**

## **Modul 7**

### **“Stack”**



**Disusun Oleh:**

**Doni Wicaksono - 21104062**

**SE-05-02**

**Dosen:**

**Yudha Islami Sulistya, S.Kom., M.Cs**

**PROGRAM STUDI S1 SOFTWARE ENGINEERING  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2024**

```
// stack.h

#ifndef STACK_H
#define STACK_H

const int STACK_CAPACITY = 15;

typedef char elementType;

struct stack {
    elementType elements[STACK_CAPACITY];
    int topIndex;
};

void initializeStack(stack &stk);
bool isEmptyStack(stack stk);
bool isStackFull(stack stk);
void pushElement(stack &stk, elementType element);
elementType popElement(stack &stk);
void displayStack(stack stk);

#endif
```

#### Penjelasan:

File ini mendefinisikan konstanta `STACK_CAPACITY` yang menentukan ukuran maksimal stack yaitu 15, lalu mendefinisikan tipe data `elementType` sebagai `char` karena stack akan menyimpan karakter. File ini juga mendefinisikan struktur `stack` yang berisi array `elements` untuk menyimpan data dan variabel `topIndex` untuk menandai posisi atas stack.

```
// stack.cpp

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

using namespace std;

void initializeStack(stack &stk) {
    stk.topIndex = 0;
}

bool isEmptyStack(stack stk) {
    return (stk.topIndex == 0);
}

bool isStackFull(stack stk) {
    return (stk.topIndex == STACK_CAPACITY);
}

void pushElement(stack &stk, elementType element) {
    if (isStackFull(stk)) {
        cout << "Stack is full! Cannot add more elements." << endl;
    } else {
        stk.elements[stk.topIndex] = element;
        stk.topIndex++;
        cout << "Element " << element << " successfully added to stack." << endl;
    }
}
```

```

// stack.cpp

elementType popElement(stack &stk) {
    if (isStackEmpty(stk)) {
        cout << "Stack is empty! No elements to remove." << endl;
        return 0;
    } else {
        stk.topIndex--;
        return stk.elements[stk.topIndex];
    }
}

void displayStack(stack stk) {
    if (isStackEmpty(stk)) {
        cout << "Stack is empty!" << endl;
    } else {
        for (int idx = stk.topIndex - 1; idx >= 0; idx--) {
            cout << stk.elements[idx] << " ";
        }
        cout << endl;
    }
}

```

#### Penjelasan:

Ada fungsi initializeStack untuk membuat stack kosong dengan mengatur topIndex ke 0, fungsi isStackEmpty untuk mengecek apakah stack kosong dengan melihat apakah topIndex = 0, fungsi isStackFull untuk mengecek apakah stack penuh dengan melihat apakah topIndex sudah sama dengan STACK\_CAPACITY. Lalu ada fungsi pushElement untuk menambah elemen ke stack dengan cara menyimpan elemen di posisi topIndex lalu menambah topIndex, fungsi popElement untuk mengambil dan menghapus elemen teratas dengan cara mengurangi topIndex lalu mengembalikan elemen di posisi tersebut, dan fungsi displayStack untuk menampilkan semua isi stack dari atas ke bawah.

```

// main.cpp

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

using namespace std;

int main() {
    stack myStack;
    initializeStack(myStack);

    int lastNIMDigit = 2;
    string wordSequence;

    switch (lastNIMDigit % 4) {
        case 0:
            wordSequence = "IFLABJAYA";
            cout << "Result:" << endl;
            break;
        case 1:
            wordSequence = "HALOBANDUNG";
            cout << "Result:" << endl;
            break;
        case 2:
            wordSequence = "PERCAYADIRI";
            cout << "Result:" << endl;
            break;
        case 3:
            wordSequence = "STRUKTURDATA";
            cout << "Result:" << endl;
            break;
        default:
            cout << "Invalid NIM last digit." << endl;
            return 0;
    }

    for (int idx = wordSequence.length() - 1; idx >= 0; idx--) {
        pushElement(myStack, wordSequence[idx]);
    }

    cout << "Isi Stack Awal: ";
    displayStack(myStack);

    int elementsToRemove = wordSequence.length() - 4;
    for (int idx = 0; idx < elementsToRemove; idx++) {
        popElement(myStack);
    }

    cout << "Isi Stack Setelah Pop: ";
    displayStack(myStack);

    return 0;
}

```

#### Penjelasan:

Program ini dimulai dengan membuat stack kosong, lalu ada variabel lastNIMDigit yang diset ke 2 dan akan menentukan kata apa yang akan dimasukkan ke stack berdasarkan sisa bagi 4 (bisa IFLABJAYA, HALOBANDUNG, PERCAYADIRI, atau STRUKTURDATA). Program akan memasukkan huruf-huruf dari kata yang terpilih ke dalam stack dari belakang ke depan menggunakan fungsi pushElement. Setelah itu program menampilkan isi stack, lalu menghapus beberapa huruf dengan fungsi

popElement sampai tersisa 4 huruf saja, dan terakhir menampilkan isi stack yang sudah dikurangi.

### Output:

```
Result:
Element I successfully added to stack.
Element R successfully added to stack.
Element I successfully added to stack.
Element D successfully added to stack.
Element A successfully added to stack.
Element Y successfully added to stack.
Element A successfully added to stack.
Element C successfully added to stack.
Element R successfully added to stack.
Element E successfully added to stack.
Element P successfully added to stack.
Isi Stack Awal: P E R C A Y A D I R I
Isi Stack Setelah Pop: D I R I
@slashedzer0 →/workspaces/STD_Doni_Wicaksono_21104062/07_Stack/TP (main) $
```