

LAPORAN PRAKTIKUM
MODUL 7
STACK



Disusun Oleh:
Prajna paramitha - 2311104016
SE 07 01

Dosen :
Yudha Islami Sulistya, S.Kom., M.Cs

PROGRAM STUDI S1 SOFTWARE ENGINEERING
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

SOAL TP

// stack.h

```
#ifndef STACK_H
#define STACK_H

// Mendefinisikan tipe data dasar untuk stack
typedef char infotype;

// Mendefinisikan struktur stack
struct stack {
    infotype info[15]; // Array untuk menyimpan data dengan ukuran 15
    int Top;           // Menunjukkan indeks teratas dari stack
};

// Deklarasi fungsi-fungsi yang akan digunakan
void createStack_2311104016(stack &S); // Membuat stack kosong
bool isEmpty_2311104016(stack S);      // Mengecek stack kosong
bool isFull_2311104016(stack S);       // Mengecek stack penuh
void push_2311104016(stack &S, infotype x); // Menambah elemen
infotype pop_2311104016(stack &S);     // Mengambil elemen teratas
void printInfo_2311104016(stack S);    // Mencetak isi stack

#endif
```

Penjelasan :

- Membuat rancangan sesuai dengan yang tertera di modul TP
- createStack : untuk membuat stack baru yang kosong
- bool isEmpty : mengecek apakah stack kosong
- bool isFull : mengecek apakah stack penuh
- void push : untuk menambahkan elemen ke stack
- infotype pop : mengambil elemen teratas dari stack
- void printInfo : menampilkan isi stack

// stack.cpp

```
#include "stack.h"
#include <iostream>
using namespace std;
//implementasi ADT stack

// Membuat stack baru yang kosong
void createStack_2311104016(stack &S) {
    S.Top = 0; // Set top ke 0 menandakan stack kosong
}

// Mengecek apakah stack kosong
bool isEmpty_2311104016(stack S) {
    if (S.Top == 0) {
        return true;
    } else {
        return false;
    }
}

// Mengecek apakah stack penuh
bool isFull_2311104016(stack S) {
    if (S.Top == 15) {
        return true;
    } else {
        return false;
    }
}

// Menambahkan elemen ke stack
void push_2311104016(stack &S, infotype x) {
    if (!isFull_2311104016(S)) { // Jika stack belum penuh
        S.Top++; // Naikkan nilai top
        S.info[S.Top] = x; // Masukkan elemen ke posisi top
    }
}

// Mengambil elemen teratas dari stack
infotype pop_2311104016(stack &S) {
    infotype x;
    x = S.info[S.Top]; // Ambil elemen teratas
    S.Top--; // Kurangi nilai top
    return x; // Kembalikan elemen yang diambil
}

// Menampilkan isi stack
void printInfo_2311104016(stack S) {
    for (int i = 1; i <= S.Top; i++) {
        cout << S.info[i]; // Cetak karakter
        if (i < S.Top) { // Jika bukan karakter terakhir
            cout << " "; // Tambahkan spasi
        }
    }
    cout << endl;
}
```

Penjelasan :

- Membuat implementasi dari ADT yang sebelumnya dibuat di file stack.h
- Void createStack {S.Top = 0} //set nilai top = 0 artinya stack kosong
- Bool isEmpty untuk mengecek apakah stack kosong atau tidak dengan mengecek top dari stacknya, yang artinya jika top stack = 0 artinya stack kosong
- Bool isFull untuk mengecek apakah stack kosong atau tidak dengan mengecek top dari stacknya, karena maksimalnya 15 maka jika banyaknya sudah 15 maka stacknya sudah penuh
- Void push untuk menambahkan elemen ke dalam stack
- Infotype pop untuk mengambil elemen teratas dari stack
- Void printinfo untuk menampilkan isi stack

// main.cpp

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

// Fungsi untuk memproses satu kasus output
void processCase_2311104016(string input, int popCount, string caseLabel) {
    stack S;
    createStack_2311104016(S); // Inisialisasi stack kosong

    // Push semua karakter ke dalam stack
    for (char c : input) {
        push_2311104016(S, c);
    }

    cout << "Digit terakhir NIM MOD 4 sisa " << caseLabel << ":" << endl;
    cout << "Output:" << endl;

    // Tampilkan kondisi awal stack
    printInfo_2311104016(S);

    // Pop sejumlah karakter sesuai kebutuhan
    for (int i = 0; i < popCount; i++) {
        pop_2311104016(S);
    }

    // Tampilkan kondisi akhir stack
    printInfo_2311104016(S);
    cout << endl; // Tambah baris kosong antara kasus
}

int main() {
    // Kasus 0: IFLABJAYA -> JAYA
    processCase_2311104016("IFLABJAYA", 7, "0");

    // Kasus 1: HALOBANDUNG -> BANDUNG
    processCase_2311104016("HALOBANDUNG", 4, "1");

    // Kasus 2: PERCAYADIRI -> DIRI
    processCase_2311104016("PERCAYADIRI", 7, "2");

    // Kasus 3: STRUKTURDATA -> DATA
    processCase_2311104016("STRUKTURDATA", 8, "3");

    return 0;
}
```

Penjelasan :

- Membuat method processcase untuk memproses satu kasus output
- Lalu memanggil createStack untuk menginialisasi stack kosong
- Lalu for char : untuk push semua karakter kedalam stack
- Lalu menampilkan digit terakhir dengan cout
- Menampilkan kondisi awal stack dengan memanggil printInfo
- For (int i=0..) untuk pop sejumlah karakter sesuai kebutuhan
- Lalu memanggil printInfo untuk menampilkan kondisi akhir stack
- Lalu membuat fungsi main yang akan dijalankan pertama kali. Dengan memanggil fungsi processCase.