

LAPORAN PRAKTIKUM

MODUL 7

STACK



Disusun Oleh:

Muhammad Fathi Rafa -2311104022

SE07A

Dosen :

Yudha Islami Prasetya, S.Kom., M.Cs.

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

INSTITUT TEKNOLOGI TELKOM

PURWOKERTO

2024

Soal Tugas Pendahuluan

Stack.h

```
#ifndef STACK_H
#define STACK_H

#include <iostream>
using namespace std;

typedef char infotype;

struct Stack {
    infotype info[15];
    int Top;
};

// Fungsi dan Prosedur untuk ADT Stack
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, int num_to_pop);

#endif
```

File berguna untuk mendefinisikan struktu data stack dan prototipe fungsi-fungsi yang terkait dalam stack. Yang pertama struct stack digunakan untuk mendefinisikan struktur stack yang beranggotakan dua anggota. Selanjutnya mendefinisikan fungsi yang ditulis.

Stack.cpp

```

#include "stack.h"

// Membuat stack baru dengan Top = 0
void createStack(Stack &S) {
    S.Top = 0;
}

// Mengecek apakah stack kosong
bool isEmpty(Stack S) {
    return (S.Top == 0);
}

// Mengecek apakah stack penuh
bool isFull(Stack S) {
    return (S.Top == 15);
}

// Menambahkan elemen ke dalam stack
void push(Stack &S, infotype x) {
    if (!isFull(S)) {
        S.Top++;
        S.info[S.Top] = x;
    } else {
        cout << "Stack penuh!" << endl;
    }
}

// Mengeluarkan elemen dari stack
infotype pop(Stack &S) {
    if (!isEmpty(S)) {
        infotype x = S.info[S.Top];
        S.Top--;
        return x;
    } else {
        cout << "Stack kosong!" << endl;
        return '\0'; // Mengembalikan karakter
    }
}

```

Pada file stack.cpp berisi semua fungsi yang akan dibuat. Yang pertama, membuat stack baru dengan Top = 0. Kedua, fungsi untuk mengecek stack kosong. Ketiga, fungsi untuk mengecek fungsi penuh. Ke-empat, fungsi untuk menambahkan elemen ke dalam stack. Ke-lima, fungsi untuk mengeluarkan elemen dari stack.

```

// Menampilkan semua elemen di stack dari bawah ke atas dan melakukan
void printInfo(Stack S, int num_to_pop) {
    // Cetak isi stack awal
    cout << "Isi stack awal: ";
    for (int i = 1; i <= S.Top; i++) {
        cout << S.info[i] << " ";
    }
    cout << endl;
    // Pop sejumlah elemen sesuai permintaan dan tampilkan sisanya
    cout << "Isi stack setelah pop: ";
    for (int i = 0; i < num_to_pop; i++) {
        pop(S);
    }
    for (int i = 1; i <= S.Top; i++) {
        cout << S.info[i] << " ";
    }
    cout << endl;
}

```

Ke-enam, fungsi untuk mencetak semua elemen dari stack dari bawah ke atas dan melakukan pop sesuai jumlah tertentu.

Main.cpp

```

#include "stack.h"

// Fungsi untuk menguji stack berdasar
void testStack(int remainder) {
    Stack S;
    createStack(S);

    string data;
    int num_to_pop;

    // Menentukan data dan jumlah ele
    if (remainder == 0) {
        data = "IFLABJAYA";
        num_to_pop = 5;
    } else if (remainder == 1) {
        data = "HALOBANDUNG";
        num_to_pop = 7;
    } else if (remainder == 2) {
        data = "PERCAYADIRI";
        num_to_pop = 8;
    } else if (remainder == 3) {
        data = "STRUKTURDATA";
        num_to_pop = 9;
    }
}

```

File main berguna sebagai eksekutor semua kode. Membuat fungsi untuk menguji operasi stack yang menerima satu parameter. Stack S digunakan menyimpan elemen stack selama proses pengujian. Lalu, memanggil fungsi createstack yang terdapat pda file stack.h. baris berikutnya, mendeklarasikan 2 variabel. Struktur if-else untuk menentukan reminder,data dan num_to_pop mana yang dipakai.

```

// Menyimpan data ke dalam stack
for (char c : data) {
    push(S, c);
}

// Cetak hasil sesuai dengan jumlah
printInfo(S, num_to_pop);
}

int main() {
    int NIM_last_digit = 2;
    int remainder = NIM_last_digit % 4;

    testStack(remainder);

    return 0;
}

```

Pada main, menetapkan nim yang ingin digunakan(untuk reminder), setelah itu memanggil fungsi teststack