

LAPORAN PRAKTIKUM
MODUL 07
STACK



Nama :

Tegar Kang Ageng Gilang (2311104018)

Dosen :

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

PROGRAM STUDI S1 REKAYASA PERANGKAT
LUNAK

FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

TP (Tugas Pendahuluan)

1. Anda diminta untuk mengimplementasikan stack dengan representasi statis, yang memanfaatkan array sebagai media penyimpanannya.

File stack.h :

```
#ifndef STACK_H
#define STACK_H

#define MAX_SIZE 15 // Ukuran maksimum stack
typedef char infotype; // Tipe data untuk elemen stack

// Struktur untuk stack
struct stack {
    infotype info[MAX_SIZE]; // Array untuk menyimpan elemen
    int Top; // Indeks teratas dari stack
};

// Prototipe fungsi
void createStack(stack &S);
bool isEmpty(stack S);
bool isFull(stack S);
void push(stack &S, infotype x);
int pop(stack &S);
void printInfo(stack S);

#endif
```

File stack.cpp :

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

// Menginisialisasi stack
void createStack(stack &S) {
    S.Top = -1; // Inisialisasi Top ke -1 agar stack kosong
}

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

// Mengecek apakah stack penuh
bool isFull(stack S) {
    return S.Top == MAX_SIZE - 1; // Top maksimal adalah MAX_SIZE - 1
}

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

// Menghapus elemen dari stack dan mengembalikannya
int pop(stack &S) {
    infotype x = '\0'; // Inisialisasi x
    if (!isEmpty(S)) {
        x = S.info[S.Top]; // Ambil elemen teratas
        S.Top--; // Mengurangi Top
    } else {
        cout << "Stack kosong, tidak bisa melakukan pop." << endl;
    }
    return x; // Mengembalikan elemen yang dihapus
}

// Menampilkan elemen stack
void printInfo(stack S) {
    if (isEmpty(S)) {
        cout << "Stack kosong." << endl;
    } else {
        cout << "Isi stack dari atas ke bawah: ";
        for (int i = S.Top; i >= 0; i--) { // Menampilkan dari Top ke bawah
            cout << S.info[i] << " ";
        }
        cout << endl;
    }
}
```

File main.cpp :

```
#include <iostream>
#include <cstring> // Untuk fungsi strlen
#include "stack.h"
#include "stack.cpp"
using namespace std;

void displayStackOutput(stack &S, const char* elements, int popCount) {
    createStack(S);

    int length = strlen(elements); // Menggunakan strlen untuk mendapatkan panjang
    // Memasukkan elemen ke dalam stack secara terbalik
    for (int i = length - 1; i >= 0; i--) {
        push(S, elements[i]);
    }

    cout << "Isi stack awal:\n";
    printInfo(S);

    // Melakukan operasi pop sesuai dengan jumlah yang ditentukan
    for (int i = 0; i < popCount; i++) {
        pop(S);
    }

    cout << "Isi stack setelah beberapa kali pop:\n";
    printInfo(S);
}

int main() {
    stack S;
    int lastDigit;

    cout << "Masukkan digit terakhir NIM: ";
    cin >> lastDigit;

    cout << "Digit terakhir NIM MOD 4 sisa " << (lastDigit % 4) << ":\n";

    // Menentukan elemen untuk dimasukkan ke stack berdasarkan hasil modulus
    switch (lastDigit % 4) {
        case 0:
            displayStackOutput(S, "IFLABJAYA", 5);
            break;
        case 1:
            displayStackOutput(S, "HALOBANDUNG", 4);
            break;
        case 2:
            displayStackOutput(S, "PERCAYADIRI", 7);
            break;
        case 3:
            displayStackOutput(S, "STRUKTURDATA", 8);
            break;
        default:
            cout << "Input tidak valid" << endl;
    }

    return 0;
}
```

Output :

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

TP git:(main) x cd "/Users/macbook15/Documents/KAMPUS/AKADEMIK/SEMESTER 3/STRUKTUR DATA/PRAKTIKUM/ 07_Stack/TP/"
" && g++ main.cpp -o main && "/Users/macbook15/Documents/KAMPUS/AKADEMIK/SEMESTER 3/STRUKTUR DATA/PRAKTIKUM/ 07_Stack/TP/"main
Masukkan digit terakhir NIM: 0
Digit terakhir NIM MOD 4 sisa 0:
Isi stack awal:
Isi stack dari atas ke bawah: I F L A B J A Y A
Isi stack setelah beberapa kali pop:
Isi stack dari atas ke bawah: J A Y A
TP git:(main) x cd "/Users/macbook15/Documents/KAMPUS/AKADEMIK/SEMESTER 3/STRUKTUR DATA/PRAKTIKUM/ 07_Stack/TP/"
" && g++ main.cpp -o main && "/Users/macbook15/Documents/KAMPUS/AKADEMIK/SEMESTER 3/STRUKTUR DATA/PRAKTIKUM/ 07_Stack/TP/"main
Masukkan digit terakhir NIM: 1
Digit terakhir NIM MOD 4 sisa 1:
Isi stack awal:
Isi stack dari atas ke bawah: H A L O B A N D U N G
Isi stack setelah beberapa kali pop:
Isi stack dari atas ke bawah: B A N D U N G
TP git:(main) x cd "/Users/macbook15/Documents/KAMPUS/AKADEMIK/SEMESTER 3/STRUKTUR DATA/PRAKTIKUM/ 07_Stack/TP/"
" && g++ main.cpp -o main && "/Users/macbook15/Documents/KAMPUS/AKADEMIK/SEMESTER 3/STRUKTUR DATA/PRAKTIKUM/ 07_Stack/TP/"main
Masukkan digit terakhir NIM: 2
Digit terakhir NIM MOD 4 sisa 2:
Isi stack awal:
Isi stack dari atas ke bawah: P E R C A Y A D I R I
Isi stack setelah beberapa kali pop:
Isi stack dari atas ke bawah: D I R I
TP git:(main) x cd "/Users/macbook15/Documents/KAMPUS/AKADEMIK/SEMESTER 3/STRUKTUR DATA/PRAKTIKUM/ 07_Stack/TP/"
" && g++ main.cpp -o main && "/Users/macbook15/Documents/KAMPUS/AKADEMIK/SEMESTER 3/STRUKTUR DATA/PRAKTIKUM/ 07_Stack/TP/"main
Masukkan digit terakhir NIM: 3
Digit terakhir NIM MOD 4 sisa 3:
Isi stack awal:
Isi stack dari atas ke bawah: S T R U K T U R D A T A
Isi stack setelah beberapa kali pop:
Isi stack dari atas ke bawah: D A T A
TP git:(main) x
```

Penjabaran :

Program di atas merupakan sebuah implementasi program stack yang tidak hanya mencakup operasi dasar seperti push, pop, dan print, tetapi juga mengintegrasikan logika untuk mengelola dan menampilkan karakter-karakter dari string sesuai dengan input yang diberikan oleh user. Program ini dirancang sedemikian rupa sehingga setiap string yang berbeda dapat dimasukkan ke dalam struktur stack berdasarkan digit terakhir dari Nomor Induk Mahasiswa (NIM) yang dimiliki user. Proses ini dilakukan dengan menghitung nilai modulus 4 dari

digit tersebut, sehingga setiap hasil modul berfungsi sebagai kriteria untuk menentukan string mana yang akan disimpan di dalam stack. Dengan pendekatan ini, user dapat melihat bagaimana karakter-karakter dari berbagai string diorganisir dan diatur dengan rapi dalam stack, sehingga meningkatkan keterbacaan dan pemahaman struktur data yang digunakan. Selain itu, fungsi print yang diimplementasikan dalam program ini memastikan bahwa user dapat dengan mudah mengakses dan meninjau isi stack, memberikan wawasan lebih lanjut tentang bagaimana data diolah dan ditangani selama eksekusi program.

