

LAPORAN PRAKTIKUM
MODUL 6
STACK



Disusun oleh:

Arjun Werdho Kumoro

NIM : 2311102009

Dosen Pengampu:

Wahyu Andi Saputra, S.Pd., M.Eng.

PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024

BAB I

TUJUAN PRAKTIKUM

1. Mampu memahami konsep stack pada struktur data dan algoritma
2. Mampu mengimplementasikan operasi-operasi pada stack
3. Mampu memecahkan permasalahan dengan solusi stack

BAB II

DASAR TEORI

Pengertian Stack

Stack adalah struktur data sederhana yang digunakan untuk menyimpan data (mirip dengan Linked Lists). Dalam tumpukan, urutan kedatangan data penting. Sebuah tumpukan piring di kafetaria adalah contoh bagus dari tumpukan. Piring ditambahkan ke tumpukan saat mereka dibersihkan dan ditempatkan di bagian atas. Ketika sebuah piring dibutuhkan, diambil dari bagian atas tumpukan. Piring pertama yang ditempatkan di tumpukan adalah yang terakhir digunakan.

Jenis-jenis Stack

Berdasarkan kemampuan menyimpan data, struktur data stack dapat dibagi menjadi 2 jenis, yaitu: register stack dan memory stack.

1. Register stack

Register stack merupakan stack yang hanya mampu menampung data dalam jumlah yang kecil. Kedalaman maksimum pada register stack cenderung dibatasi karena ukuran unit memorinya sangat kecil dibandingkan dengan memory stack.

2. Memory stack

Pada stack jenis ini, kedalaman dari stack cukup fleksibel dan mampu menangani dalam skala yang lebih besar dibandingkan jenis sebelumnya.

BAB III

GUIDED

1. GUIDED 1

SOURCE CODE

```
#include <iostream>
using namespace std;
string arrayBuku[5];
int maksimal = 5, top = 0;
bool isFull()
{
    return (top == maksimal);
}
bool isEmpty()
{
    return (top == 0);
}
void pushArrayBuku(string data)
{
    if (isFull())
    {
        cout << "Data telah penuh" << endl;
    }
    else
    {
        arrayBuku[top] = data;
        top++;
    }
}
void popArrayBuku()
{
    if (isEmpty())
    {

```

```

        cout << "Tidak ada data yang dihapus" << endl;
    }
    else
    {
        arrayBuku[top - 1] = "";
        top--;
    }
}

void peekArrayBuku(int posisi)
{
    if (isEmpty())
    {
        cout << "Tidak ada data yang bisa dilihat" << endl;
    }
    else
    {
        int index = top;
        for (int i = 1; i <= posisi; i++)
        {
            index--;
        }
        cout << "Posisi ke " << posisi << " adalah " <<
arrayBuku[index] << endl;
    }
}

int countStack()
{
    return top;
}

void changeArrayBuku(int posisi, string data)
{
    if (posisi > top)
    {
        cout << "Posisi melebihi data yang ada" << endl;
    }
}

```

```

    }
    else
    {
        int index = top;
        for (int i = 1; i <= posisi; i++)
        {
            index--;
        }
        arrayBuku[index] = data;
    }
}

void destroyArraybuku()
{
    for (int i = top; i >= 0; i--)
    {
        arrayBuku[i] = "";
    }
    top = 0;
}

void cetakArrayBuku()
{
    if (isEmpty())
    {
        cout << "Tidak ada data yang dicetak" << endl;
    }
    else
    {
        for (int i = top - 1; i >= 0; i--)
        {
            cout << arrayBuku[i] << endl;
        }
    }
}

int main()

```

```
{

    pushArrayBuku("Kalkulus");
    pushArrayBuku("Struktur Data");
    pushArrayBuku("Matematika Diskrit");
    pushArrayBuku("Dasar Multimedia");
    pushArrayBuku("Inggris");
    cetakArrayBuku();
    cout << "\n";
    cout << "Apakah data stack penuh? " << isFull() << endl;
    cout << "Apakah data stack kosong? " << isEmpty() << endl;
    peekArrayBuku(2);
    popArrayBuku();
    cout << "Banyaknya data = " << countStack() << endl;
    changeArrayBuku(2, "Bahasa Jerman");
    cetakArrayBuku();
    cout << "\n";
    destroyArraybuku();
    cout << "Jumlah data setelah dihapus: " << top << endl;
    cetakArrayBuku();
    return 0;
}
```

SCREENSHOOT PROGRAM

```
Inggris
Dasar Multimedia
Matematika Diskrit
Struktur Data
Kalkulus

Apakah data stack penuh? 1
Apakah data stack kosong? 0
Posisi ke 2 adalah Dasar Multimedia
Banyaknya data = 4
Dasar Multimedia
Bahasa Jerman
Struktur Data
Kalkulus

Jumlah data setelah dihapus: 0
Tidak ada data yang dicetak
```

DESKRIPSI PROGRAM

popArrayBuku: Menghapus buku teratas. countStack: Menghitung jumlah buku dalam stack. changeArrayBuku: Mengganti buku pada posisi kedua dari atas dengan "Bahasa Jerman". destroyArraybuku: Menghapus semua buku dalam stack.

UNGUIDED

1. UNGUIDED 1

Buatlah program untuk menentukan apakah kalimat tersebut yang diinputkan dalam program stack adalah palindrom/tidak. Palindrom kalimat yang dibaca dari depan dan belakang sama. Jelaskan bagaimana cara kerja programnya

SOURCE CODE

```
#include <iostream>
#include <stack>
#include <string>

using namespace std;

bool isPalindrome(string str) {
    stack<char> charStack;
    int length = str.length();
    int i, mid = length / 2;

    for (i = 0; i < mid; i++) {
        charStack.push(str[i]);
    }

    for (i = mid + length % 2; i < length; i++) {
        if (charStack.top() != str[i]) {
            return false;
        }
        charStack.pop();
    }

    return true;
}
```

```

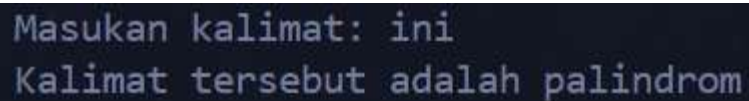
int main() {
    string input;
    cout << "Masukan kalimat: ";
    cin >> input;

    if (isPalindrome(input)) {
        cout << "Kalimat tersebut adalah palindrom";
    } else {
        cout << "Kalimat tersebut bukan palindrom";
    }

    return 0;
}

```

SCREENSHOOT PROGRAM



```

Masukan kalimat: ini
Kalimat tersebut adalah palindrom

```

DESKRIPSI PROGRAM

Memanggil fungsi isPalindrome dengan input. Menampilkan hasil apakah string adalah palindrom atau bukan berdasarkan nilai yang dikembalikan fungsi isPalindrome.

2. UNGUIDED 1

Buatlah program untuk menentukan apakah kalimat tersebut yang diinputkan dalam program stack adalah palindrom/tidak. Palindrom kalimat yang dibaca dari depan dan belakang sama. Jelaskan bagaimana cara kerja programnya

SOURCE CODE

```

#include <iostream>
#include <stack>

```

```
#include <sstream>
#include <string>
using namespace std;

// Fungsi untuk membalikkan huruf-huruf dalam setiap kata
string reverseWords(string sentence) {
    stack<char> charStack;
    stringstream ss(sentence);
    string word;
    string reversedSentence = "";

    while (ss >> word) {
        for (char c : word) {
            charStack.push(c);
        }

        while (!charStack.empty()) {
            reversedSentence += charStack.top();
            charStack.pop();
        }

        reversedSentence += ' ';
    }

    if (!reversedSentence.empty()) {
        reversedSentence.pop_back();
    }

    return reversedSentence;
}

int main() {
    string sentence;
```

```

    cout << "Masukkan kalimat : ";
    getline(cin, sentence);

    int wordCount = 0;
    stringstream ss(sentence);
    string word;
    while (ss >> word) {
        wordCount++;
    }

    if (wordCount < 3) {
        cout << "Kalimat harus terdiri dari minimal 3 kata." <<
endl;
    } else {
        string result = reverseWords(sentence);
        cout << "Data: " << result << endl;
    }

    return 0;
}

```

SCREENSHOOT PROGRAM



```

Masukkan kalimat : Arjun Werdho Kumoro
Data: nujrA ohdreW oromuK

```

DESKRIPSI PROGRAM

Program ini membaca sebuah kalimat dari input pengguna. Menghitung jumlah kata dalam kalimat tersebut. Jika jumlah kata kurang dari 3, menampilkan pesan kesalahan. Jika jumlah kata 3 atau lebih, membalikkan huruf dalam setiap kata dan menampilkan hasilnya. Fungsi `reverseWords` menggunakan `stack` untuk membalikkan urutan huruf dalam setiap kata..

BAB IV

KESIMPULAN

Setelah melakukan pembelajaran mengenai Hash Table di Bahasa Pemrograman C++ berikut poin utama yang telah dipelajari :

1. Stack adalah salah satu jenis struktur data yang menyimpan elemen-elemen data secara berurutan dengan prinsip LIFO (Last In, First Out).
2. Struktur ini sering diilustrasikan sebagai tumpukan buku, di mana buku yang terakhir ditambahkan berada di paling atas dan akan diambil pertama kali.
3. Prinsip ini berarti bahwa elemen yang terakhir kali dimasukkan (push) ke dalam stack adalah elemen yang pertama kali diambil (pop) dari stack.

DAFTAR PUSTAKA

Trivusi. (2022, 16 September) Struktur Data Stack: Pengertian, Karakteristik, dan Kegunaannya. diakses pada 20 Mei 2024 dari <https://www.trivusi.web.id/2022/07/struktur-data-stack.html>