

LAPORAN PRAKTIKUM
MODUL 6
STACK



Disusun oleh:
Aji Tri Prasetyo
NIM : 2311102064

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 sebuah kumpulan data dimana data yang diletakkan di atas data yang lain. Dengan demikian stack adalah struktur data yang menggunakan konsep LIFO (Last In First Out). Dengan demikian, elemen terakhir yang disimpan dalam stack menjadi elemen pertama yang diambil. Dalam proses komputasi, untuk meletakkan sebuah elemen pada bagian atas dari stack, maka dilakukan operasi push. Dan untuk memindahkan dari tempat yang atas tersebut, maka dilakukan operasi pop.

Penjelasan dan Algoritma:

1) Pada awalnya dibuat sebuah struct 'struct STACK', kemudian struct stack dijadikan menjadi sebuah tipe data dari variabel stackbaru. Dan beberapa prototype fungsi diantaranya :

- void clear()
- void print()
- void pop()
- void push(float dta)
- bool isempty()
- bool isfull()

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

Program ini mengimplementasikan operasi dasar pada stack, termasuk push, pop, peek, cetak, dan fungsi tambahan seperti mengubah elemen tertentu dan menghapus seluruh elemen dalam stack. Dengan menggunakan stack, data buku disimpan dengan cara LIFO (Last In, First Out).

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;
}
```

```

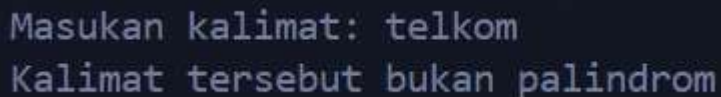
int main() {
    string input;
    cout << "Masukan kalimat: ";
    cin >> input;

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

    return 0;
}

```

SCREENSHOOT PROGRAM



```

Masukan kalimat: telkom
Kalimat tersebut bukan palindrom

```

DESKRIPSI PROGRAM

Program ini membaca string dari input pengguna dan memeriksa apakah string tersebut adalah palindrom. Fungsi `isPalindrome` menggunakan stack untuk membandingkan karakter dari bagian pertama dan bagian kedua string.

2. UNGUIDED 1

Buatlah program untuk melakukan pembalikan terhadap kalimat menggunakan stack dengan minimal 3 kata. Jelaskan output program dan source codenya beserta operasi/fungsi yang dibuat?

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 : ";
```

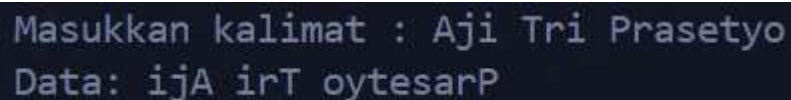
```
getline(cin, sentence);

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 : Aji Tri Prasetyo
Data: iJA irT oytesarP

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.

BAB IV

KESIMPULAN

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

1. Pemrograman Rekursif: Mengelola fungsi rekursif menggunakan stack panggilan.
2. Pembalikan String/Kata: Membalikkan urutan karakter dalam string atau urutan kata dalam kalimat.
3. Navigasi Browser: Menyimpan riwayat halaman yang dikunjungi.

DAFTAR PUSTAKA

Unknown. (2014, 26 April) Stack pada C++. diakses pada 20 Mei 2024 dari <https://www.nblognlife.com/2014/04/stack-pada-c.html>