

LAPORAN PRAKTIKUM
MODUL 6
STACK



Disusun oleh:
Reza Alvonzo
NIM : 2311102026

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

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. Definisi: Sebuah tumpukan adalah daftar terurut di mana penyisipan dan penghapusan dilakukan di satu ujung, disebut atas. Elemen terakhir yang dimasukkan adalah yang pertama dihapus. Oleh karena itu, disebut daftar Last in First out (LIFO). Operasi pada stack melibatkan beberapa fungsi dasar yang dapat dilakukan pada struktur data ini. Berikut adalah beberapa operasi umum pada stack: a. Push (Masukkan): Menambahkan elemen ke dalam tumpukan pada posisi paling atas atau ujung. b. Pop (Keluarkan): Menghapus elemen dari posisi paling atas atau ujung tumpukan. c. Top (Atas): Mendapatkan nilai atau melihat elemen teratas pada tumpukan tanpa menghapusnya. d. IsEmpty (Kosong): Memeriksa apakah tumpukan kosong atau tidak. e. IsFull (Penuh): Memeriksa apakah tumpukan penuh atau tidak (terutama pada implementasi tumpukan dengan kapasitas terbatas). f. Size (Ukuran): Mengembalikan jumlah elemen yang ada dalam tumpukan. g. Peek (Lihat): Melihat nilai atau elemen pada posisi tertentu dalam tumpukan tanpa menghapusnya. h. Clear (Hapus Semua): Mengosongkan atau menghapus semua elemen dari tumpukan. i. Search (Cari): Mencari keberadaan elemen tertentu dalam tumpukan.

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

pushArrayBuku(string data): Fungsi ini menambahkan elemen baru ke tumpukan. Jika tumpukan penuh, fungsi ini menampilkan pesan "Data telah penuh". Jika tidak, elemen baru ditambahkan ke indeks top dan nilai top diincrement.

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;
}
```

```

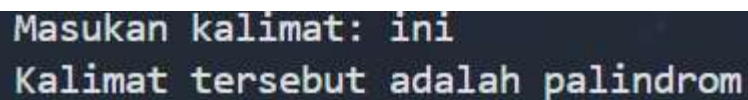
int main() {
    string input;
    cout << "Masukan kalimat: ";
    cin >> input;

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

    return 0;
}

```

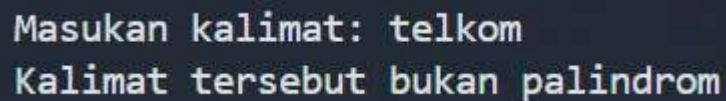
SCREENSHOOT PROGRAM



```

Masukan kalimat: ini
Kalimat tersebut adalah palindrom

```



```

Masukan kalimat: telkom
Kalimat tersebut bukan palindrom

```

DESKRIPSI PROGRAM

`bool isPalindrome(string str)`: Fungsi ini menerima sebuah string `str` sebagai input dan mengembalikan nilai `true` jika `str` adalah palindrom, dan `false` jika bukan.

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 += ' ';
    }

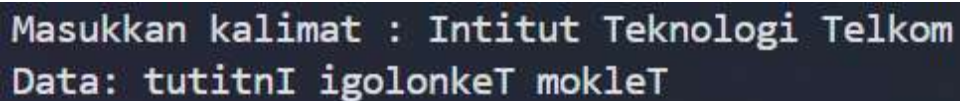
    if (!reversedSentence.empty()) {
        reversedSentence.pop_back();
    }

    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

A screenshot of a terminal window with a dark background. It shows the program's output: "Masukkan kalimat : Intitut Teknologi Telkom" on the first line and "Data: tutitnI igolonkeT mokleT" on the second line. The text is in a light-colored, monospaced font.

```
Masukkan kalimat : Intitut Teknologi Telkom  
Data: tutitnI igolonkeT mokleT
```

DESKRIPSI PROGRAM

String reverseWords(string sentence): Fungsi ini menerima sebuah string sentence sebagai input dan mengembalikan string yang berisi kalimat dengan setiap kata di dalamnya dibalik urutan hurufnya.

BAB IV

KESIMPULAN

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

1. Undo/Redo: Stack dapat digunakan untuk menyimpan status program sebelumnya sehingga pengguna dapat membatalkan atau mengulang tindakan yang telah dilakukan.
2. Efisiensi ruang: Stack hanya membutuhkan memori untuk menyimpan elemen yang sedang aktif, sehingga lebih hemat ruang dibandingkan struktur data lain seperti array.
3. Memori yang terbuang: Jika stack tidak dikosongkan secara teratur, memori yang digunakan untuk menyimpan elemen yang tidak lagi dibutuhkan akan terbuang.

DAFTAR PUSTAKA

Sigit, N. (2023, 21 Februari) Contoh Membuat Program dengan Konsep Stack C++. diakses pada 21 Mei 2024 dari <https://pemburukode.com/contoh-program-konsep-stack-cpp/>