

**LAPORAN PRAKTIKUM
MODUL VI
STACK**



Disusun oleh:

Raihan Ramadhan

NIM: 2311102040

Dosen Pengampu:

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

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
PURWOKERTO
2024**

BAB I

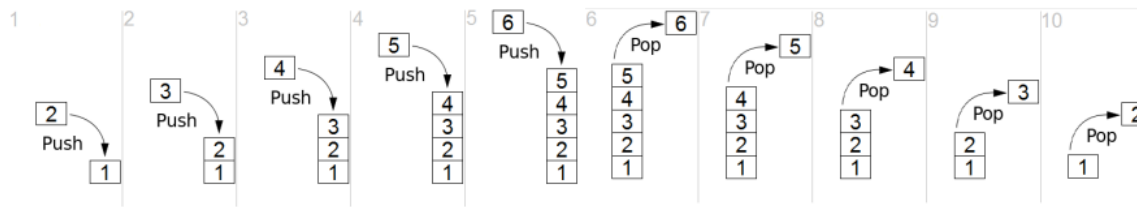
TUJUAN PRAKTIKUM

Praktikum ini bertujuan untuk memastikan bahwa mahasiswa memiliki pemahaman yang baik tentang apa itu stack, bagaimana cara kerjanya, dan mengapa penting dalam pemrograman. Mahasiswa diharapkan dapat menjelaskan dengan jelas apa itu stack dan bagaimana ia bekerja sebagai bagian dari struktur data. Praktikum ini memberikan kesempatan kepada mahasiswa untuk mengimplementasikan stack dalam bahasa pemrograman c++. Melalui latihan konkret, mahasiswa dapat belajar bagaimana mendefinisikan dan menggunakan stack dalam konteks nyata. Mahasiswa mungkin akan diberikan situasi atau masalah yang memerlukan penggunaan stack, dan mereka akan belajar bagaimana cara menguji keefektifan solusi mereka menggunakan stack.

BAB II

DASAR TEORI

Stack adalah suatu tumpukan dari benda. Konsep utamanya adalah LIFO (Last In First Out), benda yang terakhir masuk dalam stack akan menjadi benda pertama yang dikeluarkan dari stack.



Ada dua cara penerapan prinsip stack, yakni dengan array dan linked list. Setidaknya stack haruslah memiliki operasi-operasi sebagai berikut.

- Push Untuk menambahkan item pada tumpukan paling atas
- Pop Untuk mengambil item teratas
- Clear Untuk mengosongkan stack
- IsEmpty Untuk memeriksa apakah stack kosong
- IsFull Untuk memeriksa apakah stack sudah penuh
- Retreive Untuk mendapatkan nilai dari item teratas

Sesuai dengan sifat stack, pengambilan / penghapusan di elemen dalam stack harus dimulai dari elemen teratas. Operasi-operasi pada Stack dengan Array.

- IsFull Fungsi ini memeriksa apakah stack yang ada sudah penuh. Stack penuh jika puncak stack terdapat tepat di bawah jumlah maksimum yang dapat ditampung stack atau dengan kata lain $Top = MAX_STACK - 1$.
- Push Fungsi ini menambahkan sebuah elemen ke dalam stack dan tidak bisa dilakukan lagi jika stack sudah penuh.
- IsEmpty Fungsi menentukan apakah stack kosong atau tidak. Tanda bahwa stack kosong adalah Top bernilai kurang dari nol.
- Pop Fungsi ini mengambil elemen teratas dari stack dengan syarat stack tidak boleh kosong.
- Clear Fungsi ini mengosongkan stack dengan cara mengeset Top dengan -1. Jika Top bernilai kurang dari nol maka stack dianggap kosong.
- Retreive Fungsi ini untuk melihat nilai yang berada pada posisi tumpukan teratas

Stack dengan Single Linked List Operasi-operasi untuk Stack dengan Linked List :

- IsEmpty Fungsi memeriksa apakah stack yang adamasih kosong.
- Push Fungsi memasukkan elemen baru ke dalam stack. Push di sini mirip dengan insert dalam single linked list biasa.
- Pop Fungsi ini mengeluarkan elemen teratas dari stack

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;
}

```

Screenshot Output

```
PS D:\Matkul\SEMESTER 2\PRAKTIKUM STRUKDAT\Modul 6> .\Guided1_Stack.cpp -o Guided1_Stack } ; if ($?) { .\Guided1_Stack
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
PS D:\Matkul\SEMESTER 2\PRAKTIKUM STRUKDAT\Modul 6>
```

Deskripsi Program

Program tersebut mengimplementasikan struktur data stack menggunakan array dalam bahasa C++. Stack ini memiliki kapasitas maksimal lima elemen dengan fungsi untuk menambah (push), menghapus (pop), dan melihat (peek) elemen berdasarkan prinsip LIFO (Last In, First Out). Fungsi tambahan mencakup pemeriksaan apakah stack penuh atau kosong, menghitung jumlah elemen, mengubah data pada posisi tertentu, dan menghapus seluruh elemen dalam stack. Program utama mendemonstrasikan penggunaan fungsi-fungsi ini dengan menambah beberapa buku ke stack, mencetak isi stack, memodifikasi data, dan menghapus elemen. Pesan-pesan khusus dicetak untuk setiap operasi yang berhasil atau gagal, memberikan umpan balik kepada pengguna.

BAB IV

UNGUIDED

1. Unguided 1

Source Code

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

using namespace std;

bool cekPalindrom(const string &kalimat)
{
    stack<char> tumpukan;

    for (char ch : kalimat)
    {
        if (isalnum(ch))
        {
            tumpukan.push(tolower(ch));
        }
    }

    for (char ch : kalimat)
    {
        if (isalnum(ch))
        {
            if (tumpukan.top() != tolower(ch))
            {
                return false;
            }
            tumpukan.pop();
        }
    }

    return true;
}
```



```

int main()
{
    string kalimat;
    cout << "Masukkan Kalimat: ";
    getline(cin, kalimat);

    if (cekPalindrom(kalimat))
    {
        cout << "Kalimat tersebut adalah palindrom\n";
    }
    else
    {
        cout << "Kalimat tersebut bukan palindrom\n";
    }

    return 0;
}

```

Screenshot Program

```

PS D:\Matkul\SEMESTER 2\PRAKTIKUM STRUKDAT\Modul 6> cd
ded_1.cpp -o Unguided_1 } ; if ($?) { .\Unguided_1 }
Masukkan Kalimat: ini
Kalimat tersebut adalah palindrom
PS D:\Matkul\SEMESTER 2\PRAKTIKUM STRUKDAT\Modul 6> 

```

```

PS D:\Matkul\SEMESTER 2\PRAKTIKUM STRUKDAT\Modul 6> cd
ded_1.cpp -o Unguided_1 } ; if ($?) { .\Unguided_1 }
Masukkan Kalimat: ittp
Kalimat tersebut bukan palindrom
PS D:\Matkul\SEMESTER 2\PRAKTIKUM STRUKDAT\Modul 6> 

```

Deskripsi Program

Program yang diberikan adalah aplikasi sederhana dalam bahasa C++ yang bertujuan untuk menentukan apakah sebuah kalimat yang dimasukkan oleh pengguna adalah palindrom atau tidak. Algoritma yang digunakan dalam program ini mengandalkan struktur data stack untuk membalikkan kalimat dan membandingkannya dengan kalimat aslinya. Pertama-tama, program meminta pengguna untuk memasukkan kalimat melalui konsol. Selanjutnya, program membersihkan kalimat dari spasi dan karakter non-alfanumerik serta mengubah semua huruf menjadi huruf kecil untuk memudahkan proses perbandingan. Setelah itu, setiap karakter alfanumerik dari kalimat dimasukkan ke dalam stack secara terbalik. Akhirnya, program membandingkan karakter-karakter dalam kalimat dengan karakter-karakter dalam stack untuk menentukan apakah kalimat tersebut adalah palindrom atau bukan. Jika kalimat tersebut adalah palindrom, program akan mencetak pesan "Kalimat tersebut adalah palindrom", dan jika tidak, program akan mencetak pesan "Kalimat tersebut bukan palindrom".

2. Unguided 2

Source Code

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

using namespace std;

int main()
{
    stack<char> s;
    string kalimat, hasil;

    cout << "Masukkan Kata (input/kalimat): ";
    getline(cin, kalimat);

    for (char c : kalimat)
    {
        s.push(c);
    }

    while (!s.empty())
    {
        hasil += s.top();
        s.pop();
    }

    cout << "Kata Terbalik: " << hasil << endl;

    return 0;
}
```

Screenshot Output

```
PS D:\Matkul\SEMESTER 2\PRAKTIKUM STRUKDAT\Modul 6> cd
ded_2.cpp -o Unguided_2 } ; if ($?) { .\Unguided_2 }
Masukkan Kata: Raihan Ramadhan
Kata Terbalik: nahdamaR nahiaR
PS D:\Matkul\SEMESTER 2\PRAKTIKUM STRUKDAT\Modul 6> |
```

Deskripsi Program

Program ini meminta pengguna untuk memasukkan sebuah kata atau kalimat melalui input. Setelah itu, program membaca setiap karakter dari input dan menyimpannya ke dalam sebuah stack. Selanjutnya, program mengosongkan stack satu per satu dan menyimpan karakter-karakter tersebut ke dalam variabel hasil untuk mendapatkan kata atau kalimat yang sudah terbalik. Akhirnya, program mencetak kata atau kalimat terbalik tersebut. Program ini merupakan contoh sederhana dari penggunaan stack untuk membalikkan urutan dari sebuah teks.

BAB V

KESIMPULAN

Ketiga program tersebut memperlihatkan beragam penerapan konsep stack dalam konteks yang berbeda. Program pertama memanfaatkan struktur data stack untuk mengatur daftar judul buku dengan operasi dasar seperti penambahan, penghapusan, dan pencetakan. Hal ini mencerminkan penggunaan stack dalam manajemen daftar. Sementara itu, program kedua menggunakan stack untuk melakukan pengecekan palindrom pada kata atau kalimat, menunjukkan bagaimana stack digunakan dalam algoritma pengecekan khusus. Program ketiga, di sisi lain, mengimplementasikan stack untuk membalikkan urutan karakter dalam sebuah kata atau kalimat, memberikan contoh penggunaan stack dalam manipulasi sederhana pada data. Semuanya menunjukkan fleksibilitas dan kegunaan struktur data stack dalam berbagai situasi pemrograman.

DAFTAR PUSTAKA

Asisten Praktikum “MODUL VI STACK”. Learning Management System, 20 Maret 2024.

Muliono Rizki, S.Kom, M.Kom “ Pemrograman C++ ALGORITMA & STRUKTUR DATA”. Chrome, 19 Mei 2024. <https://abdulkadir.blog.uma.ac.id/wp-content/uploads/sites/365/2019/02/Modul-Praktikum-Algoritma-Struktur-Data-Merge.pdf>