

LAPORAN PRAKTIKUM

MODUL VI

STACK (TUMPUKAN)



Disusun Oleh:

Muhammad Raafi Al Hafiidh

2311102070

Dosen:

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

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

BAB I

TUJUAN PRAKTIKUM

- Mahasiswa mampu menjelaskan definisi dan konsep dari Stack.
- Mahasiswa mampu menerapkan Stack kedalam pemrograman.

BAB II

DASAR TEORI

Pengertian Stack (Tumpukan)

Stack, atau tumpukan, adalah struktur data abstrak yang menyimpan elemen data dalam urutan LIFO (Last In, First Out). Artinya, elemen terakhir yang dimasukkan ke dalam stack adalah elemen pertama yang akan dikeluarkan. Stack sering diilustrasikan seperti tumpukan piring, di mana piring terakhir yang diletakkan di atas adalah piring pertama yang akan diambil.

Karakteristik Stack

- **LIFO (Last In, First Out):** Elemen terakhir yang dimasukkan ke dalam stack adalah elemen pertama yang akan dikeluarkan.
- **Akses Terbatas:** Elemen hanya dapat diakses dari atas stack.
- **Operasi Dasar:**
 - **Push:** Menambahkan elemen baru ke atas stack.
 - **Pop:** Menghapus dan mengembalikan elemen teratas dari stack.
 - **Peek:** Mendapatkan nilai elemen teratas tanpa menghapusnya.
 - **Empty:** Memeriksa apakah stack kosong.

Representasi Stack

Stack dapat direpresentasikan dengan berbagai cara, seperti array, linked list, atau struktur data khusus yang dirancang untuk stack. Implementasi stack yang umum adalah dengan menggunakan array, di mana elemen stack disimpan dalam array dan indeks top menunjuk ke elemen teratas.

Aplikasi Stack

Stack memiliki banyak aplikasi dalam ilmu komputer, termasuk:

- **Ekspresi Matematika:** Stack digunakan untuk mengevaluasi ekspresi matematika dengan menggunakan notasi infix dan postfix.
- **Kompilasi dan Interpretasi Bahasa Pemrograman:** Stack digunakan untuk menyimpan variabel lokal, parameter fungsi, dan informasi kontrol selama kompilasi dan interpretasi bahasa pemrograman.
- **Undo/Redo:** Stack digunakan untuk menyimpan status program sebelumnya sehingga pengguna dapat membatalkan atau mengulang tindakan mereka.
- **Algoritma Pencocokan Pola:** Stack digunakan untuk mencocokkan pola dalam teks, seperti kurung kuadrat dan tanda kurung.

- **Browser Web:** Stack digunakan untuk menyimpan riwayat halaman yang dikunjungi pengguna dalam browser web.

Kelebihan Stack

- **Sederhana:** Stack memiliki struktur yang sederhana dan mudah dipahami.
- **Efisien:** Operasi dasar stack (push, pop, peek) dapat dilakukan dengan waktu $O(1)$.
- **Serbaguna:** Stack memiliki banyak aplikasi dalam berbagai bidang ilmu komputer.

Kekurangan Stack

- **Akses Terbatas:** Elemen hanya dapat diakses dari atas stack, sehingga sulit untuk mengakses elemen di tengah stack.
- **Tidak Urut:** Elemen dalam stack tidak memiliki urutan tertentu, sehingga sulit untuk mencari elemen tertentu dalam stack.

BAB III

LATIHAN DAN TUGAS

A. GUIDED

1. Guided

Source Code

```
#include <iostream>
using namespace std;
string arrayBuku[5];
int maksimal = 5, top = 0;
bool isFull()
{
    if (top == maksimal)
    {
        return true;
    }
    else
    {
        return false;
    }
}
bool isEmpty()
{
    if (top == 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}
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--)

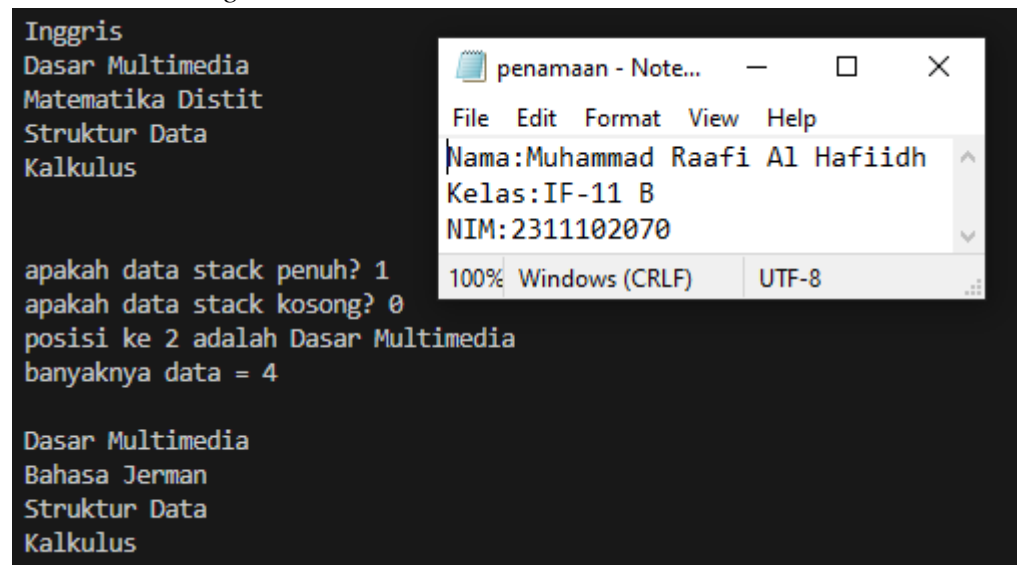
```

```

    {
        arrayBuku[i] = "";
    }
    top = 0;
}
void cetakArrayBuku()
{
    if (isEmpty())
    {
        cout << "tidak ada data yang dicetak" << endl;
    }
    else
    {
        for (int i = top; i >= 0; i--)
        {
            cout << arrayBuku[i] << endl;
        }
    }
}
int main()
{
    pushArrayBuku("Kalkulus");
    pushArrayBuku("Struktur Data");
    pushArrayBuku("Matematika Distit");
    pushArrayBuku("Dasar Multimedia");
    pushArrayBuku("Inggris");
    cetakArrayBuku();
    cout << "\n"
         << endl;
    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"
         << endl;
    destroyArraybuku();
    cout << top;
    cetakArrayBuku();
}

```

Screenshoot Program



Deskripsi Program

Kode di atas menggunakan stack pada proses fungsinya, pertama program akan memasukkan beberapa nama buku dimulai dari kalkulus sampai dasar multimedia, karena stack bersifat LIFO maka kalkulus akan berada dibawah sedangkan dasar multimedia akan berada diatas.

B. UNGUIDED

1. Unguided 1

Buatlah Program untuk melakukan pembalikan terhadap kalimat dengan menggunakan stack.

Contoh :

Kalimat : Telkom Purwokerto

Hasil : otrekowruP mokleT

```
Masukkan Kata Telkom Purwokerto  
Datastack Array :  
Data : otrekowruP mokleT
```

Source Code

```
#include <iostream>  
#include <stack>  
#include <string>  
  
using namespace std;  
  
string reverseLetters(const string &sentence)  
{
```



```

    stack<char> letterStack;
    string reversedSentence;
    string word;

    for (char c : sentence)
    {
        if (c == ' ')
        {
            while (!letterStack.empty())
            {
                reversedSentence += letterStack.top();
                letterStack.pop();
            }
            reversedSentence += ' ';
        }
        else
        {
            letterStack.push(c);
        }
    }
    while (!letterStack.empty())
    {
        reversedSentence += letterStack.top();
        letterStack.pop();
    }
    return reversedSentence;
}

bool isPalindrome(const string &sentence)
{
    stack<char> charStack;
    // Push each character of the word into the stack
    for (char c : sentence)
    {
        charStack.push(c);
    }
    // Compare each character from the stack with the
word
    for (char c : sentence)
    {
        if (c != charStack.top())
        {
            return false;
        }
        charStack.pop();
    }
    return true;
}

```

```

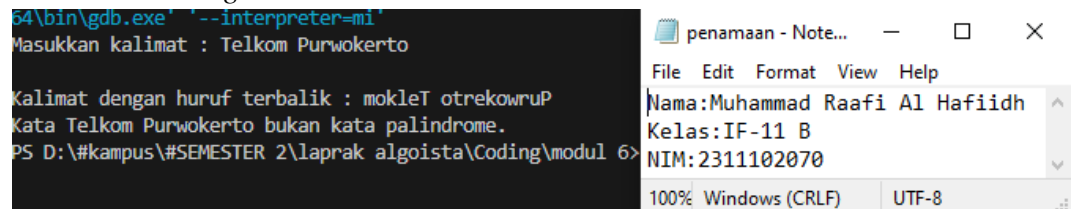
int main()
{
    string sentence;
    cout << "Masukkan kalimat : ";
    getline(cin, sentence);
    cout << endl;
    string reversed = reverseLetters(sentence);
    cout << "Kalimat dengan huruf terbalik : " <<
    reversed << endl;

    if (isPalindrome(sentence))
    {
        cout << "Kata " << sentence << " adalah kata
        palindrome." << endl;
    }
    else
    {
        cout << "Kata " << sentence << " bukan kata
        palindrome." << endl;
    }

    return 0;
}

```

Screenshoot Program



Deskripsi Program

Kode diatas menggunakan stack pada proses fungsinya, pertama program akan memasukkan sebuah kata atau kalimat dan nantinya kata atau kalimat tersebut akan dibalik posisi dan tulisannya melalui proses stack.

2. Unguided 2

Dari soal no. 1 buatlah program untuk menentukan apakah kalimat tersebut yang diinputkan dalam program stack adalah polindrom/tidak. Polindrom kalimat yang dibaca dari depan dan belakang sama

Contoh :

Kalimat : ini

Kalimat tersebut adalah polindrom

Kalimat : telkom

Kalimat tersebut adalah bukan polindrom

```
Masukan Kalimat : ini
Kalimat tersebut adalah : Palindrom
```

Algoritma :

1. Mulai
2. Masukkan kata
3. Hitung jumlah
4. Masukkan kedalam stack (push)
5. Bandingkan elemen 1 dalam stack dengan elemen terakhir (pop)
6. Perbandingan dilakukan berulang sebanyak jumlah huruf
7. Jika huruf yang dibandingkan semuanya sama, kata tersebut adalah polindrom
8. Selesai

Source Code

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

using namespace std;

string reverseLetters(const string &sentence)
{
    stack<char> letterStack;
    string reversedSentence;
    string word;

    for (char c : sentence)
    {
        if (c == ' ')
        {
            while (!letterStack.empty())
            {
                reversedSentence += letterStack.top();
                letterStack.pop();
            }
            reversedSentence += ' ';
        }
        else
        {
            letterStack.push(c);
        }
    }
}
```

```

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

    bool isPalindrome(const string &sentence)
    {
        stack<char> charStack;
        // Push each character of the word into the stack
        for (char c : sentence)
        {
            charStack.push(c);
        }
        // Compare each character from the stack with the
        word
        for (char c : sentence)
        {
            if (c != charStack.top())
            {
                return false;
            }
            charStack.pop();
        }
        return true;
    }

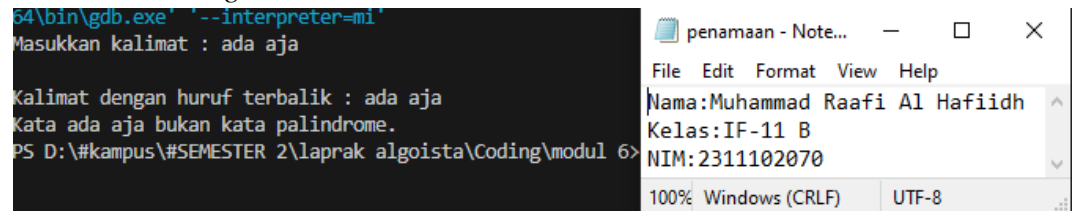
    int main()
    {
        string sentence;
        cout << "Masukkan kalimat : ";
        getline(cin, sentence);
        cout << endl;
        string reversed = reverseLetters(sentence);
        cout << "Kalimat dengan huruf terbalik : " <<
        reversed << endl;

        if (isPalindrome(sentence))
        {
            cout << "Kata " << sentence << " adalah kata
            palindrome." << endl;
        }
        else
        {
            cout << "Kata " << sentence << " bukan kata
            palindrome." << endl;
        }
    }

```

```
    return 0;  
}
```

Screenshoot Program



Deskripsi Program

Kode diatas menggunakan stack pada proses fungsinya, pertama program akan memasukkan sebuah kata atau kalimat dan nantinya kata atau kalimat tersebut akan dicek apakah kata atau kalimat tersebut merupakan Palindrom (kata yang apabila dibalik bunyi atau maknanya sama) atau tidak melalui proses stack.

BAB IV

KESIMPULAN

Kesimpulan yang dapat diambil dari laporan praktikum dengan “*Modul 6 : STACK (TUMPUKAN)*” adalah bahwa praktikum ini bertujuan untuk mempelajari fungsi dari penggunaan Stack dalam pemrograman. Praktisi jadi bisa memahami penggunaan dari konsep tersebut dan bagaimana kegunaan dari konsep tersebut dalam program yang dibuat, serta setelah melakukan pratikum, mahasiswa mampu menerapkan konsep tersebut didalam program yang mereka buat. Seperti contohnya praktisi membuat sebuah program untuk menginputkan sebuah kata atau kalimat yang nantinya kata atau kalimat tersebut akan dibalik posisi dengan penyusunan katanya, serta mengecek apakah kata tersebut polindrom atau tidak menggunakan konsep Stack.

Reverensi

"Stack Machines: The Emergence of a New Paradigm" by Donald Knuth