

**LAPORAN PRAKTIKUM STRUKTUR DATA DAN  
PEMROGRAGAMAN**

**MODUL 6  
STACK**



**Disusun oleh :**

Rafa Aldhino Fatin

2311102023

IF-11-A

**Dosen Pengampu :**

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

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

**2023**

# **BAB I**

## **TUJUAN PRAKTIKUM**

- a. Mahasiswa mampu memahami konsep stack pada struktur data dan algoritma
- b. Mahasiswa mampu mengimplementasikan operasi-operasi pada stack
- c. Mahasiswa mampu memecahkan permasalahan dengan solusi stack

## **BAB II**

### **DASAR TEORI**

#### **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 stack, maka dilakukan operasi push. Dan untuk memindahkan dari tempat yang atas tersebut, maka dilakukan operasi pop. Berikut ini adalah operator-operator atau nama method yang biasa digunakan dalam pemrograman algoritma stack.

Berikut adalah beberapa operasi dasar stack :

- PUSH: penyisipan (Memasukkan elemen).
- POP: penghapusan (Mengeluarkan elemen puncak).
- IsEmpty: operator yang memeriksa apakah stack kosong.
- IsFull: operator yang memeriksa apakah stack penuh.
- Clear: operator untuk menghapus semua elemen didalam stack.

## BAB III GUIDED

### 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();
}

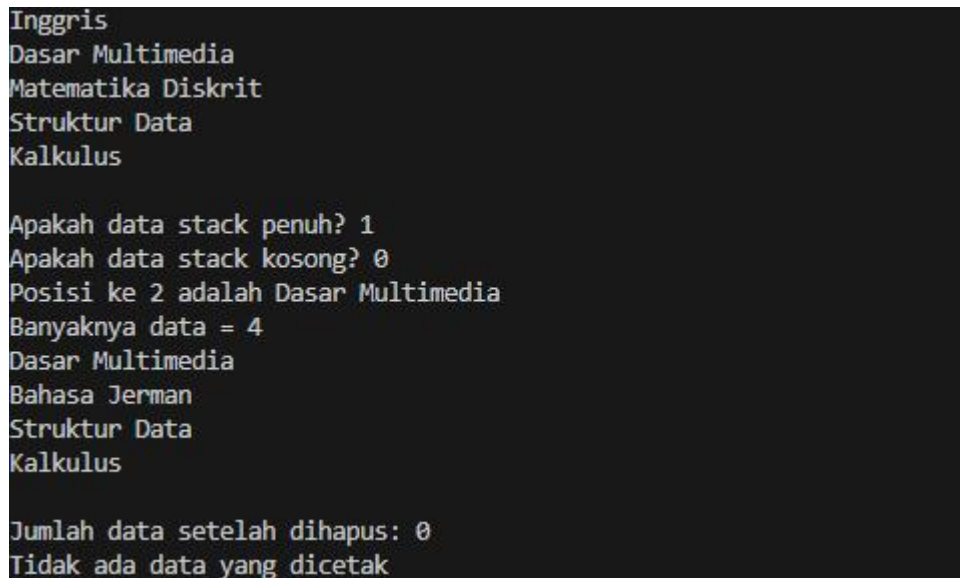
```

```

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 struktur data stack menggunakan array dengan kapasitas 5 elemen, yang bekerja berdasarkan prinsip Last-In, First-Out (LIFO). Program ini memiliki beberapa fungsi: untuk menambahkan elemen (push), menghapus elemen teratas (pop), melihat elemen di posisi tertentu (peek), menghitung jumlah elemen dalam stack, serta mengubah dan mengosongkan stack. Selain itu, program ini juga memeriksa apakah stack dalam keadaan penuh atau kosong. Misalnya, program dapat menambahkan lima elemen, mencetaknya, menghapus satu elemen, mengganti elemen di posisi tertentu, dan akhirnya mengosongkan seluruh stack. Fungsi-fungsi ini menunjukkan operasi dasar yang dapat dilakukan pada stack.

## BAB III

### UNGUIDED

#### UnGuided 1

##### Source code

```
#include <iostream>
#include <stack>
#include <string>
using namespace std;
bool SystemPalindrom2311102023(const string &input)
{
    stack<char> charCharacters;
    for (char c : input)
    {
        charCharacters.push(c);
    }
    for (char c : input)
    {
        if (c != charCharacters.top())
        {
            return false;
        }
        charCharacters.pop();
    }
    return true;
}
int main()
{
    string input;
    cout << "Masukan Kalimat: ";
    getline(cin, input);
    if (SystemPalindrom2311102023(input))
    {
        cout << "Kalimat tersebut adalah Palindrom" << endl;
    }
    else
    {
        cout << "Kalimat tersebut bukan Palindrom" << endl;
    }
    return 0;
}
```

## SCREENSHOOT PROGRAM

```
Masukan Kalimat: katak
Kalimat tersebut adalah Palindrom
PS E:\KULIAH\SOURCE CODE\SEMESTER 2\praktikum struktur data\pertemuan 7 stack> cd
guided }
Masukan Kalimat: aldhino
Kalimat tersebut bukan Palindrom
PS E:\KULIAH\SOURCE CODE\SEMESTER 2\praktikum struktur data\pertemuan 7 stack> |
```

## DESKRIPSI PROGRAM

Kode C++ ini memeriksa apakah sebuah string merupakan palindrom menggunakan struktur data stack. Fungsi `SystemPalindrom2311102023` menerima sebuah string sebagai argumen dan mengembalikan nilai boolean yang menunjukkan apakah string tersebut adalah palindrom. Fungsi ini pertama-tama menambahkan setiap karakter dari string ke dalam stack `charCharacters`. Kemudian, fungsi membandingkan setiap karakter dalam string asli dengan karakter yang diambil dari puncak stack (top) secara berurutan. Jika ditemukan perbedaan, fungsi mengembalikan `false`. Jika semua karakter cocok, fungsi mengembalikan `true`, menunjukkan bahwa string tersebut adalah palindrom. Dalam fungsi `main`, program meminta pengguna untuk memasukkan sebuah kalimat, memanggil fungsi `palindrom`, dan mencetak hasil apakah kalimat tersebut adalah palindrom atau tidak.



## UnGuided 2

### Source code

```
#include <iostream>
#include <stack>
#include <string>
using namespace std;
int main()
{
    string input;
    cout << "Masukkan kalimat (minimal 3 kata): ";
    getline(cin, input);
    stack<char> s2311102023;
    for (char c : input)
    {
        s2311102023.push(c);
    }
    string reversed;
    while (!s2311102023.empty())
    {
        char c = s2311102023.top();
        s2311102023.pop();
        reversed += c;
    }
    size_t start = 0;
    size_t end = 0;
    for (size_t i = 0; i < reversed.length(); i++)
    {
        if (reversed[i] == ' ')
        {
            end = i;
            string word = reversed.substr(start, end - start);
            cout << word << ' ';
            start = i + 1;
        }
    }

    string lastWord = reversed.substr(start);
    cout << lastWord << endl;
    return 0;
}
```

### SCREENSHOOT PROGRAM

```
Masukkan kalimat (minimal 3 kata): rafa aldhino fatin
nitaf onihdla afar
PS E:\KULIAH\SOURCE CODE\SEMESTER 2\praktikum struktur
```

## **DESKRIPSI PROGRAM**

Kode C++ ini membalik urutan karakter dalam sebuah string yang diinput , lalu menampilkan setiap kata dalam urutan aslinya setelah membalik karakter string tersebut. Program dimulai dengan meminta pengguna untuk memasukkan sebuah kalimat minimal tiga kata. Setiap karakter dalam kalimat dimasukkan ke dalam stack s2311102023. Kemudian, karakter-karakter tersebut diambil dari stack satu per satu untuk membentuk string baru reversed yang merupakan hasil pembalikan karakter dari string input. program memproses string reversed untuk memisahkan dan mencetak setiap kata dalam urutan aslinya dari string input yang sudah terbalik. Kata terakhir ditangani secara khusus karena tidak diikuti oleh spasi. Hasil akhirnya adalah tampilan kata-kata dalam urutan aslinya, tetapi dengan karakter yang telah dibalik dalam string input.

## **BAB V**

### **KESIMPULAN**

Kode C++ ini membalik urutan karakter dalam sebuah string yang diinput lalu menampilkan setiap kata dalam urutan aslinya setelah membalik karakter string tersebut. Program dimulai dengan yang di input untuk memasukkan sebuah kalimat minimal tiga kata. Setiap karakter dalam kalimat dimasukkan ke dalam stack . Kemudian, karakter-karakter tersebut diambil dari stack satu per satu untuk membentuk string baru ,yang merupakan hasil pembalikan karakter dari string input. Setelah itu, program memproses string . untuk memisahkan dan mencetak setiap kata dalam urutan aslinya dari string input yang sudah terbalik, dengan mengidentifikasi spasi sebagai pemisah kata. Hasil akhirnya adalah tampilan kata-kata dalam urutan aslinya, tetapi dengan karakter yang telah dibalik dalam string input.

### **DAFTAR PUSTAKA**

Karumanchi, N. (2016). Data Structures and algorithms made easy: Concepts, problems, Interview Questions. CareerMonk Publications.

<https://www.geeksforgeeks.org/stack-data-structure/>

[https://en.wikipedia.org/wiki/Stack\\_\(abstract\\_data\\_type\)](https://en.wikipedia.org/wiki/Stack_(abstract_data_type))