

**LAPORAN PRAKTIKUM STRUKTUR DATA DAN  
PEMROGRAMAN  
MODDUL VI  
“STUCK”**



**Disusun oleh :**

Baharuddin Barkah Pratama

2311102321

IF-11-A

**Dosen Pengampu :**

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

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

## **BAB I**

### **A. TUJUAN PRAKTIKUM**

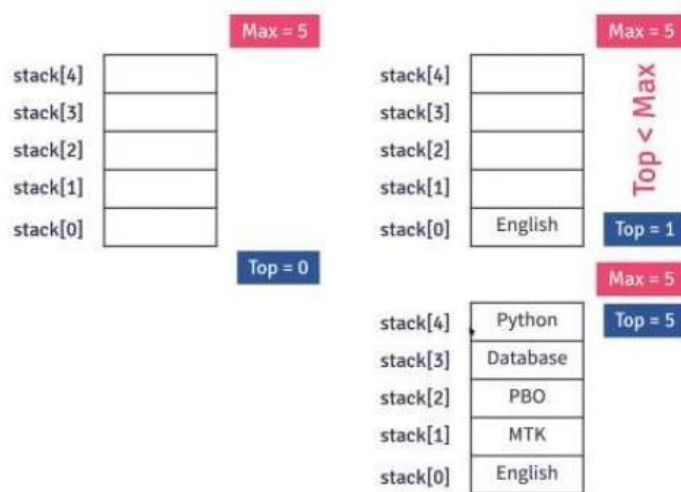
- a. Mampu memahami konsep stack pada struktur data dan algoritma
- b. Mampu mengimplementasikan operasi-operasi pada stack
- c. Mampu memecahkan permasalahan dengan solusi stack

## BAB II

### B. 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:

- Push (Masukkan):** Menambahkan elemen ke dalam tumpukan pada posisi paling atas atau ujung.
- Pop (Keluarkan):** Menghapus elemen dari posisi paling atas atau ujung tumpukan.
- Top (Atas):** Mendapatkan nilai atau melihat elemen teratas pada tumpukan tanpa menghapusnya.
- IsEmpty (Kosong):** Memeriksa apakah tumpukan kosong atau tidak.
- IsFull (Penuh):** Memeriksa apakah tumpukan penuh atau tidak (terutama pada implementasi tumpukan dengan kapasitas terbatas).
- Size (Ukuran):** Mengembalikan jumlah elemen yang ada dalam tumpukan.
- Peek (Lihat):** Melihat nilai atau elemen pada posisi tertentu dalam tumpukan tanpa menghapusnya.
- Clear (Hapus Semua):** Mengosongkan atau menghapus semua elemen dari tumpukan.
- Search (Cari):** Mencari keberadaan elemen tertentu dalam tumpukan.

## BAB III

### GUIDED

```
#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
PS C:\Users\skyzo>
```

## DESKRIPSI PROGRAM

Program tersebut mengimplementasikan stack dengan kapasitas maksimal lima elemen yang disimpan dalam array `arrayBuku`. Fungsi-fungsinya meliputi `isFull` untuk memeriksa apakah stack penuh, `isEmpty` untuk memeriksa apakah stack kosong, `pushArrayBuku` untuk menambahkan elemen ke dalam stack, `popArrayBuku` untuk menghapus elemen dari stack, `peekArrayBuku` untuk melihat elemen pada posisi tertentu tanpa menghapusnya, `countStack` untuk menghitung jumlah elemen dalam stack, `changeArrayBuku` untuk mengganti elemen pada posisi tertentu, `destroyArraybuku` untuk menghapus semua elemen dari stack, dan `cetakArrayBuku` untuk mencetak semua elemen dalam stack. Dalam program utama, beberapa buku ditambahkan ke dalam stack, isi stack dicetak, dilakukan pemeriksaan apakah stack penuh atau kosong, elemen pada posisi tertentu dilihat, elemen dihapus dari stack, elemen diganti, isi stack dicetak kembali, dan akhirnya semua elemen dihapus sebelum mencetak isi stack yang kosong.

## BAB IV

### UNGUIDED

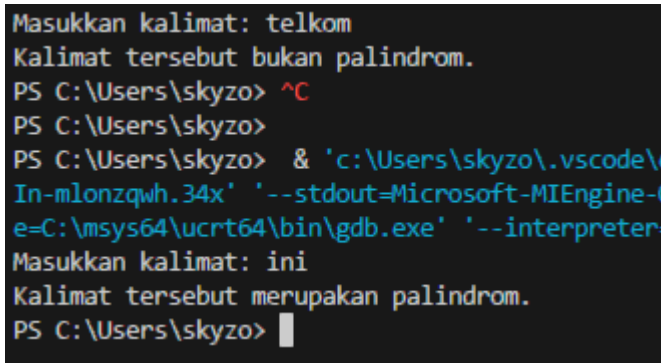
1. Buatlah program untuk menentukan apakah kalimat tersebut yang diinputkan dalam program stack adalah palindrom/tidak. Palindrom kalimat yang dibaca dari depan dan belakang sama. Jelaskan bagaimana cara kerja programnya.

#### SOURCE CODE

```
#include <iostream>
#include <stack>
#include <string>
#include <algorithm>
using namespace std;
bool isPalindrome(string str)
{
    stack<char> charStack;
    string originalStr = str;
    // Menghapus spasi dan tanda baca dari kalimat
    str.erase(remove_if(str.begin(), str.end(), ::isspace), str.end());
    str.erase(remove_if(str.begin(), str.end(), ::ispunct), str.end());
    // Memasukkan setiap karakter ke dalam stack
    for (char c : str)
    {
        charStack.push(c);
    }
    // Membandingkan karakter pada stack dengan karakter asli dari
    belakang
    for (char c : originalStr)
    {
        if (isspace(c) || ispunct(c))
        {
            continue; // Lewati spasi dan tanda baca
        }
        if (c != charStack.top())
        {
            return false; // Tidak palindrom
        }
        charStack.pop();
    }
    return true; // Palindrom
}

int main()
{
    string kalimat;
    cout << "Masukkan kalimat: ";
    getline(cin, kalimat);
    if (isPalindrome(kalimat))
    {
        cout << "Kalimat tersebut merupakan palindrom." << endl;
    }
    else
    {
        cout << "Kalimat tersebut bukan palindrom." << endl;
    }
    return 0;
}
```

## SCREENSHOOT PROGRAM



```
Masukkan kalimat: telkom
Kalimat tersebut bukan palindrom.
PS C:\Users\skyzo> ^C
PS C:\Users\skyzo>
PS C:\Users\skyzo> & 'c:\Users\skyzo\.vscode\In-mlonzqwh.34x' '--stdout=Microsoft-MIEngine-
e=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter
Masukkan kalimat: ini
Kalimat tersebut merupakan palindrom.
PS C:\Users\skyzo> |
```

## DESKRIPSI PROGRAM

Fungsi `isPalindrome` menggunakan stack untuk membalikkan urutan karakter dari kalimat yang telah dibersihkan dari spasi dan tanda baca. Pertama, kalimat asli disimpan dalam `originalStr`, kemudian kalimat tersebut dibersihkan dari spasi dan tanda baca, dan setiap karakter yang tersisa dimasukkan ke dalam stack. Selanjutnya, karakter-karakter dalam kalimat asli dibandingkan satu per satu dengan karakter di puncak stack, sambil mengabaikan spasi dan tanda baca. Jika semua karakter sesuai, fungsi mengembalikan nilai `true`, menunjukkan bahwa kalimat adalah palindrom. Program utama meminta input kalimat dari pengguna, memanggil fungsi `isPalindrome`, dan menampilkan apakah kalimat tersebut merupakan palindrom atau bukan.

## UNGUIDED 2

2. 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 <string>
using namespace std;

void reverseSentence(string sentence) {
    stack<char> charStack;
    string word = "";

    cout << "Data : ";

    for (char c : sentence) {
        if (c == ' ') {
            // Proses kata yang ada di stack
            while (!charStack.empty()) {
                cout << charStack.top();
                charStack.pop();
            }
            cout << " "; // Tambahkan spasi setelah kata
        } else {
            charStack.push(c);
        }
    }
}
```



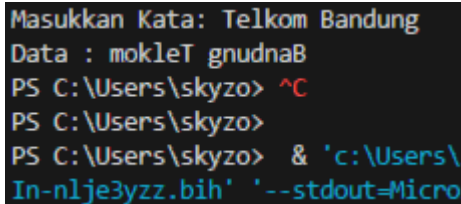
```

        // Proses kata terakhir setelah loop
        while (!charStack.empty()) {
            cout << charStack.top();
            charStack.pop();
        }
        cout << endl;
    }

    int main() {
        string kalimat;
        cout << "Masukkan Kata: ";
        getline(cin, kalimat);
        reverseSentence(kalimat);
        return 0;
    }

```

## SCREENSHOOT PROGRAM



```

Masukkan Kata: Telkom Bandung
Data : mokleT gnudnaB
PS C:\Users\skyzo> ^C
PS C:\Users\skyzo>
PS C:\Users\skyzo> & 'c:\Users\
In-nlje3yzz.bih' '--stdout=Micro

```

## DESKRIPSI PROGRAM

Program ini bertujuan untuk membalikkan setiap kata dalam sebuah kalimat, sementara urutan kata-kata dalam kalimat tetap dipertahankan. Fungsi `reverseSentence` menerima input berupa string `sentence` dan menggunakan stack untuk membalikkan karakter-karakter dalam setiap kata. Saat menemukan spasi, program memproses semua karakter yang ada dalam stack (mewakili satu kata), membalikkan urutannya, dan mencetaknya diikuti oleh spasi. Setelah loop selesai, kata terakhir yang tersisa dalam stack juga diproses dan dicetak. Program utama meminta pengguna untuk memasukkan sebuah kalimat, lalu memanggil fungsi `reverseSentence` untuk membalikkan setiap kata dalam kalimat tersebut dan mencetak hasilnya. Program ini memberikan output di mana setiap kata dalam kalimat dibalik tetapi urutan kata tetap sama.

## **BAB V**

### **KESIMPULAN**

Stack adalah struktur data yang penting dalam pemrograman, mengikuti prinsip Last In First Out (LIFO), di mana elemen terakhir yang dimasukkan adalah yang pertama kali dihapus. Operasi dasar pada stack, seperti Push untuk menambahkan elemen, Pop untuk menghapus elemen, dan Top untuk melihat elemen teratas tanpa menghapusnya, memungkinkan pengelolaan data dengan efisien. Selain itu, operasi lain seperti IsEmpty, IsFull, Size, Peek, Clear, dan Search memberikan fleksibilitas tambahan dalam penggunaan stack. Pemahaman yang baik tentang stack dan operasi-operasinya penting dalam pemrograman untuk mengimplementasikan algoritma dengan efektif dan memecahkan berbagai masalah dengan cepat dan efisien.

## **BAB VI**

### **DAFTAR PUSTAKA**

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