

**LAPORAN PRAKTIKUM STRUKTUR  
DATA DAN ALGORITMA**

**MODUL 6  
STACK**



**DISUSUN OLEH:**

**NAMA : SYARIEF RENDI ADITYA ANTONIUS**

**NIM : 2311102072**

**S1 IF-11-B**

**DOSEN:**

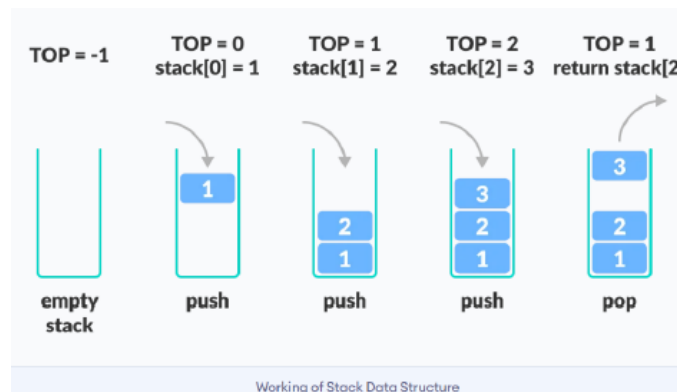
**Wahyu Andi Saputra, S.Pd., M.Eng.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
2024**

## A. 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.

- 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.

## B. Guided

### GUIDED :

#### 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 Diskirt");
    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";

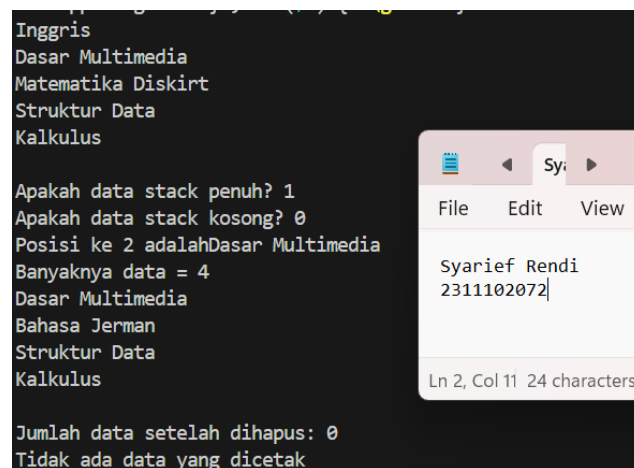
    destroyArraybuku();
    cout << "Jumlah data setelah dihapus: " << top << endl;

    cetakArrayBuku();

    return 0;
} //2311102072

```

## SCREENSHOT OUTPUT



## DESKRIPSI PROGRAM

Deskripsi untuk program di atas, sebuah array arrayBuku digunakan untuk merepresentasikan tumpukan (stack) yang menyimpan data berupa judul buku. Program ini memiliki beberapa fungsi untuk mengelola tumpukan, termasuk menambahkan data ke tumpukan (push), menghapus data dari tumpukan (pop), melihat data tertentu dalam tumpukan (peek), menghitung jumlah elemen dalam tumpukan (countStack), mengganti data pada posisi tertentu dalam tumpukan (change), mengosongkan tumpukan (destroy), dan mencetak seluruh isi tumpukan (cetak).

## C. UNGUIDED

### 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.

Contoh:

Kalimat : ini

Kalimat tersebut adalah polindrom

Kalimat : telkom

Kalimat tersebut adalah bukan polindrom

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

### SOURCE CODE

```
#include <iostream>
#include <string>
using namespace std;

const int maksimal = 100;
char stack[maksimal];
int top = 0;

bool isFull(){
    return (top == maksimal);
}

bool isEmpty(){
    return (top == 0);
}

void push(char data){
    if (!isFull()){
        stack[top] = data;
        top++;
    }
}

char pop(){
    if (!isEmpty()){
        top--;
        return stack[top];
    }
    return '\0';
}
```

```

}

bool isPalindrome(string str){
    int length = str.length();

    for (int i = 0; i < length; i++){
        push(str[i]);
    }

    for (int i = 0; i < length; i++){
        if (str[i] != pop()){
            return false;
        }
    }
    return true;
}

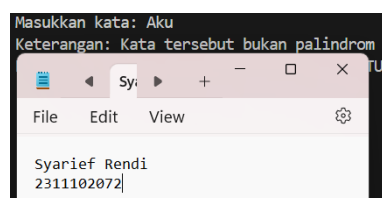
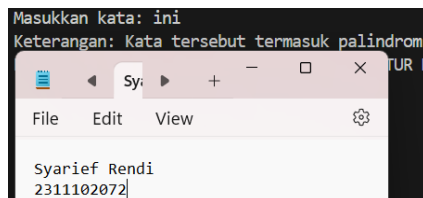
int main(){
    string input;
    cout << "Masukkan kata: ";
    cin >> input;

    if (isPalindrome(input)){
        cout << "Keterangan: Kata tersebut termasuk palindrom"
<< endl;
    } else {
        cout << "Keterangan: Kata tersebut bukan palindrom" <<
endl;
    }

    return 0;
} //2311102072

```

## SCREENSHOT OUTPUT





## **DESKRIPSI PROGRAM**

Program diatas, memeriksa apakah sebuah kata yang diinputkan adalah palindrom menggunakan struktur data tumpukan (stack). Tumpukan diimplementasikan menggunakan sebuah array yang menyimpan karakter-karakter dari kata yang diinputkan. Fungsi utama dalam program ini meliputi menambahkan karakter ke tumpukan (push), menghapus dan mengembalikan karakter dari tumpukan (pop), serta mengecek apakah tumpukan penuh atau kosong. Melalui kombinasi fungsi-fungsi ini, program dapat membalikkan urutan karakter dalam kata dan membandingkannya untuk menentukan apakah kata tersebut sama saat dibaca dari depan maupun belakang, yang merupakan definisi dari palindrom.

Algoritma bekerja dengan pertama-tama mengambil input kata dari pengguna dan mendorong setiap karakternya ke dalam tumpukan satu per satu. Setelah semua karakter masuk ke tumpukan, algoritma kemudian memulai fase perbandingan, di mana karakter-karakter diambil dari tumpukan (mulai dari yang terakhir dimasukkan) dan dibandingkan dengan karakter yang sesuai di posisi awal kata. Jika semua karakter yang diambil dari tumpukan cocok dengan karakter pada posisi yang sesuai dalam kata asli, maka kata tersebut adalah palindrom. Jika ada ketidakcocokan pada satu karakter saja, kata tersebut dinyatakan bukan palindrom. Hasilnya kemudian dicetak ke layar, menyatakan apakah kata yang diinputkan adalah palindrom atau tidak.

## UNGUIDED 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?

Contoh

Kalimat : Telkom Purwokerto

Hasil : otrekowruP mokleT

Masukkan Kata Telkom Purwokerto

Datastack Array :

Data : otrekowruP mokleT

## SOURCE CODE

```
#include <iostream>
#include <string>
using namespace std;

const int maksimal = 100;
char stack[maksimal];
int top = 0;

bool isFull() {
    return (top == maksimal);
}

bool isEmpty() {
    return (top == 0);
}

void push(char data) {
    if (!isFull()) {
        stack[top] = data;
        top++;
    }
}

char pop() {
    if (!isEmpty()) {
        top--;
        return stack[top];
    }
    return '\0';
}

string reverseString(string str) {
    int length = str.length();
```

```

        for (int i = 0; i < length; i++) {
            push(str[i]);
        }

        string reversed = "";
        while (!isEmpty()) {
            reversed += pop();
        }

        return reversed;
    }

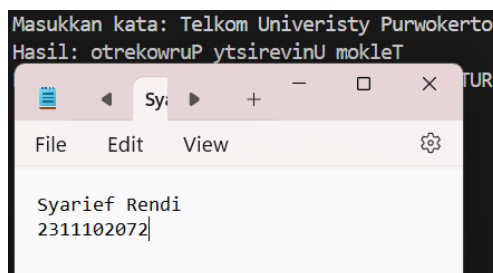
int main() {
    string input;
    cout << "Masukkan kata: ";
    getline(cin, input);

    string reversed = reverseString(input);
    cout << "Hasil: " << reversed << endl;

    return 0;
} //2311102072

```

## SCREENSHOT OUTPUT



## DESKRIPSI PROGRAM

Program diatas, membalikkan urutan karakter dari sebuah kalimat yang diinputkan oleh pengguna menggunakan struktur data tumpukan (stack). Implementasi tumpukan dilakukan menggunakan sebuah array yang menyimpan setiap karakter dari kalimat. Setelah semua karakter dari kalimat dimasukkan ke dalam tumpukan, program kemudian mengeluarkan setiap karakter dari tumpukan satu per satu, sehingga membalik urutan karakter kalimat tersebut. Hasil akhirnya adalah kalimat dengan urutan karakter yang terbalik, yang kemudian ditampilkan ke layar.

Algoritma dimulai dengan menerima input kalimat dari pengguna dan menyimpan setiap karakter dari kalimat tersebut ke dalam tumpukan

menggunakan fungsi `push(char data)`. Setelah seluruh karakter dimasukkan ke dalam tumpukan, algoritma memulai fase pembalikan dengan mengeluarkan setiap karakter dari tumpukan menggunakan fungsi `pop()`. Karena tumpukan bersifat LIFO (Last-In-First-Out), karakter pertama yang dimasukkan akan menjadi yang terakhir keluar, sehingga urutan karakter menjadi terbalik. Karakter-karakter yang dikeluarkan dari tumpukan kemudian disusun kembali menjadi sebuah string yang mewakili kalimat terbalik, dan hasil ini dicetak ke layar sebagai output akhir.

#### **D. Kesimpulan**

Stack adalah struktur data yang menggunakan prinsip Last In, First Out (LIFO), di mana elemen terakhir yang dimasukkan adalah yang pertama dikeluarkan. Contohnya seperti tumpukan piring: piring baru ditambahkan di atas dan diambil dari atas. Operasi dasar pada stack meliputi push untuk menambah elemen, pop untuk menghapus elemen teratas, top untuk melihat elemen teratas tanpa menghapusnya, isEmpty untuk memeriksa apakah stack kosong, isFull untuk memeriksa apakah stack penuh, size untuk jumlah elemen, peek untuk melihat elemen tertentu tanpa menghapusnya, clear untuk mengosongkan stack, dan search untuk mencari elemen tertentu. Stack mempermudah pengelolaan data dengan urutan yang ketat.

#### **E. Referensi**

[1] Asisten Praktikum, "Modul Stack", GoogleDrive, 2024.

[2] <https://www.trivusi.web.id/2022/07/struktur-data-stack.html>

Diakses pada 15 Mei 2024

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

[4] Budi Raharjo. 2015. Pemrograman C++. Bandung: Informatika.