

**LAPORAN PRAKTIKUM**  
**STRUKTUR DATA**  
**Modul 7**  
**“STACK”**



**Disusun Oleh:**  
**MUHAMMAD RALFI - 2211104054**  
**SE-07-2**

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

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY**  
**PURWOKERTO**  
**2024**

## 1. Tujuan

- Memahami konsep stack
- Mengimplementasikan stack dengan menggunakan representasi pointer pada tabel.
- Mampu memecahkan permasalahan dengan solusi stack.

## 2. Landasan Teori

### Stack

Stack, atau dalam Bahasa Indonesia disebut tumpukan, adalah struktur data linier yang menerapkan prinsip *Last In First Out (LIFO)*. Prinsip ini berarti elemen yang terakhir dimasukkan akan menjadi elemen pertama yang diambil. Contohnya dapat ditemukan dalam fitur undo-redo, atau dalam tumpukan piring di restoran. Dalam implementasinya, stack direpresentasikan sebagai struktur data terurut dengan dua operasi utama:

- Push*: menambahkan elemen ke dalam tumpukan.
- Pop*: mengeluarkan elemen dari tumpukan, dimulai dari elemen yang berada di posisi paling atas.

Prinsip LIFO memungkinkan stack untuk mengakses data secara efisien, menjadikannya salah satu struktur data penting dalam pengembangan perangkat lunak dan pemrograman komputer.

## 3. Guided

- Membuat stack

```
#include <iostream>
#define MAX 100
using namespace std;

class stack{
private:
    int top;
    int arr[MAX];

public:
    stack() {top = -1;}

    bool isFull() {return top == MAX -1;}
    bool isEmpty() {return top == -1;}

    void push(int x){
        if(isFull()){
            cout << "Stack Overflow\n";
            return;
        }
        arr[++top] = x;
    }

    void pop() {
        if(isEmpty()){
            cout << "Stack Underflow\n";
            return;
        }
        top--;
    }
}
```

```

int peek(){
    if(!isEmpty()){
        return arr[top];
    }
    cout << "Stack is Empty\n";
    return -1;
}
void display(){
    if(isEmpty()){
        cout << "Stack is Empty\n";
        return;
    }
    for (int i = top; i >= 0; i--){
        cout << arr[i] << " ";
    }
    cout << "\n";
}
};

int main() {
    stack s;
    s.push(10);
    s.push(20);
    s.push(40);

    cout << "Stack Element: ";
    s.display();

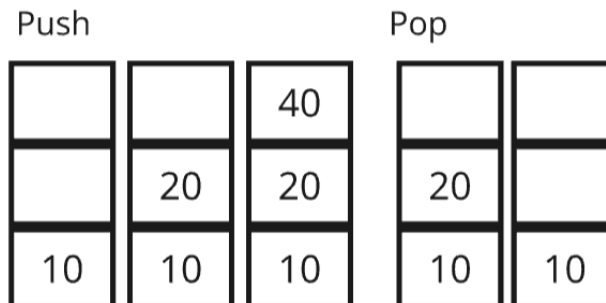
    cout << "Top element " << s.peek() << "\n";

    s.pop();
    s.pop();
    cout << "After popping, stack elements: ";
    s.display();

    return 0;
}

```

Jadi penerapan stack adalah First In Last Out (FILO) dan Last In First Out (LIFO).



Output:

```
D:\STD_Muhammad_Ralfi_2211104054\07_Stack\Guided\gd07\bin\
Stack Element: 40 20 10
Top element 40
After popping, stack elements: 10
Process returned 0 (0x0)   execution time : 0.197 s
Press any key to continue.
```

b. Membuat stack menggunakan pointer

```
// stack menggunakan pointer
#include <iostream>
using namespace std;

class Node {
public:
    int data;
    Node* next;
    Node(int value) {
        data = value;
        next = nullptr;
    }
};

class stack {
private:
    Node* top;
public:
    stack() {
        top = nullptr;
    }

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

    void push(int x) {
        Node* newNode = new Node(x);
        newNode->next = top;
        top = newNode;
    }

    void pop() {
        if(isEmpty()) {
            cout << "Stack Underflow\n";
            return;
        }
        Node* temp = top;
        top = top->next;
        delete temp;
    }

    int peek() {
        if(isEmpty()) {
            cout << "Stack is empty\n";
            return -1;
        }
    }
}
```

```

    return top->data;
}

void display() {
    if(isEmpty()) {
        cout << "Stack is empty\n";
        return;
    }
    Node* current = top;
    while (current) {
        cout << current->data << " ";
        current = current->next;
    }
    cout << "\n";
}

};

int main() {
    stack s;
    s.push(10);
    s.push(20);
    s.push(40);

    cout << "Stack Elements: ";
    s.display();

    cout << "Top element: " << s.peek() << "\n";

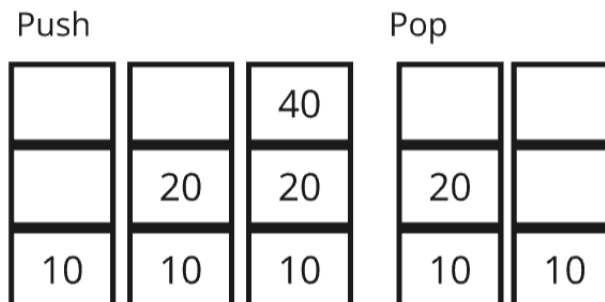
    s.pop();
    s.pop();

    cout << "After popping, stack elements: ";
    s.display();

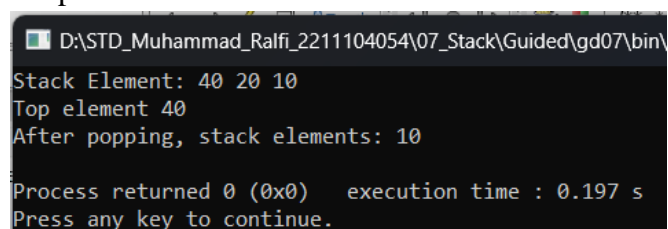
    return 0;
}

```

Jadi penerapan stack adalah First In Last Out (FILO) dan Last In First Out (LIFO).



Output:



```

D:\STD_Muhammad_Ralfi_2211104054\07_Stack\Guided\gd07\bin\
Stack Element: 40 20 10
Top element 40
After popping, stack elements: 10
Process returned 0 (0x0)   execution time : 0.197 s
Press any key to continue.

```

#### 4. Unguided

- a. Buatlah program untuk menentukan apakah kalimat tersebut yang diinputkan dalam program stack adalah polindrom/tidak. Polindrom kalimat yang dibaca dari depan dan belakang sama. Jelaskan juga bagaimana cara kerja programnya secara singkat.

Code:

```
#include <bits/stdc++.h>
using namespace std;

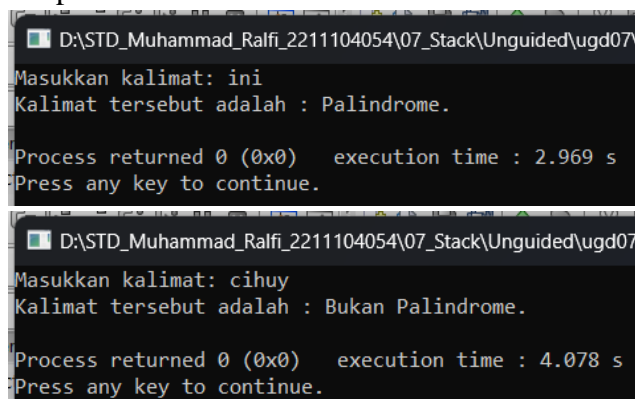
void isPalindrome(string str) {
    string rev = str;
    reverse(rev.begin(), rev.end());
    if (str == rev)
        cout << "Palindrome." << endl;
    else
        cout << "Bukan Palindrome." << endl;
}

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

    return 0;
}
```

Penjelasan code : Code ini untuk mengecek apakah kata yg dimasukkan oleh pengguna merupakan kalimat palindrome atau bukan (kalimat dibaca dari depan atau belakang tetap sama), jadi program mengecek apakah string itu sama dengan reverse nya, jika iya maka kalimat berikut adalah palindrome jika tidak maka bukan.

Output:



```
D:\STD_Muhammad_Ralfi_2211104054\07_Stack\Unguided\ugd07\
Masukkan kalimat: ini
Kalimat tersebut adalah : Palindrome.
Process returned 0 (0x0)   execution time : 2.969 s
Press any key to continue.

D:\STD_Muhammad_Ralfi_2211104054\07_Stack\Unguided\ugd07\
Masukkan kalimat: cihuy
Kalimat tersebut adalah : Bukan Palindrome.
Process returned 0 (0x0)   execution time : 4.078 s
Press any key to continue.
```

- b. Buatlah program untuk melakukan pembalikan terhadap kalimat menggunakan stack dengan minimal 3 kata, jelaskan output program nya beserta code dan fungsinya.

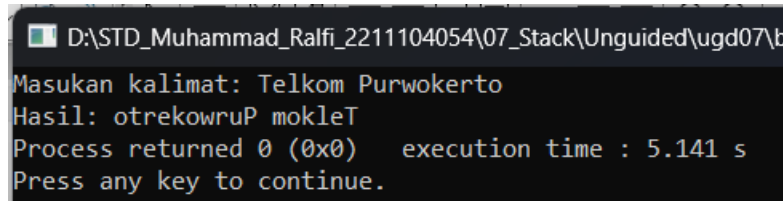
Code:

```
#include <algorithm>
#include<iostream>
#include<string>
```

```
using namespace std;
int main(){
    string inputString;
    cout << "Masukan kalimat: ";
    getline(cin, inputString);
    reverse(inputString.begin(), inputString.end());
    cout<<"Hasil: "<<inputString;
    return 0;
}
```

Penjelasan code: Program diatas merupakan code program yang digunakan untuk melakukan reverse atau membalikkan kalimat yang diinputkan oleh pengguna, getline() digunakan untuk mengecek inputan string dan spasi.

Output:



```
D:\STD_Muhammad_Ralfi_2211104054\07_Stack\Unguided\ugd07\b
Masukan kalimat: Telkom Purwokerto
Hasil: otrekowruP mokleT
Process returned 0 (0x0)   execution time : 5.141 s
Press any key to continue.
```

## 5. Kesimpulan

Stack adalah struktur data linier yang menggunakan prinsip Last In First Out (LIFO), di mana elemen terakhir yang dimasukkan menjadi yang pertama diambil. Dengan dua operasi utama, push untuk menambahkan elemen dan pop untuk menghapus elemen, stack memungkinkan akses data yang efisien. Struktur data ini penting dalam berbagai aplikasi pemrograman, seperti fitur undo-redo dan pengelolaan data berbasis tumpukan.