

**LAPORAN PRAKTIKUM STRUKTUR  
DATA DAN ALGORITMA**

**MODUL VI  
STACK**



**Disusun Oleh :**

**NAMA : FAISAL KHOIRUDDIN**

**NIM : 2311102046**

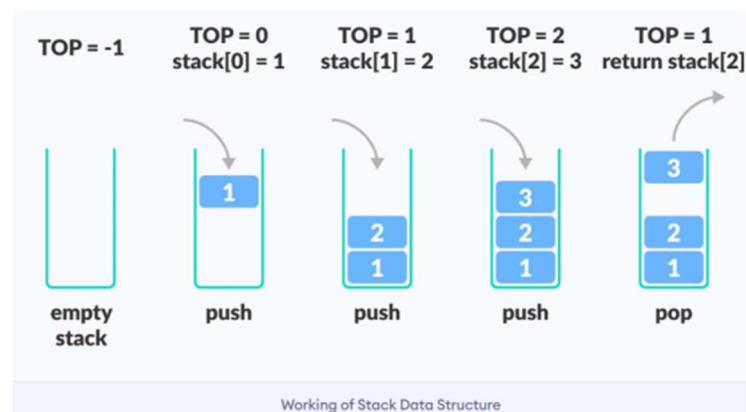
**Dosen**

**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). Konsep Stack menggunakan Metode LIFO ( Last In First Out ). 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.

- d. IsEmpty (Kosong): Memeriksa apakah tumpukan kosong atau tidak. Praktikum Struktur Data dan Algoritma 2
- e. IsFull (Penuh): Memeriksa apakah tumpukan penuh atau tidak (terutama pada implementasi tumpukan dengan kapasitas terbatas). Size (Ukuran): Mengembalikan jumlah elemen yang ada dalam tumpukan.
- f. Peek (Lihat): Melihat nilai atau elemen pada posisi tertentu dalam tumpukan tanpa menghapusnya.
- g. Clear (Hapus Semua): Mengosongkan atau menghapus semua elemen dari tumpukan.
- h. Search (Cari): Mencari keberadaan elemen tertentu dalam tumpukan.

## Guided

### 1. 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;
}

```

Screenshots Output

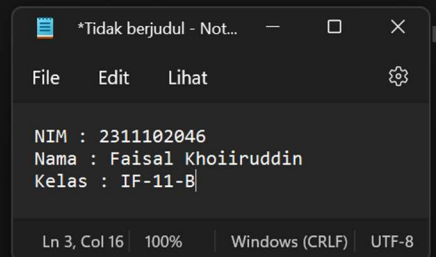
```
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vs
code.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Micro
soft-MIEngine-In-ez3gqwt3.u00' '--stdout=Microsoft-MIEngine-Out-av1lyqm5.iky' '--stderr
=Microsoft-MIEngine-Error-umz4llak.og5' '--pid=Microsoft-MIEngine-Pid-hwffl01x.i02' '--d
bgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'

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

0tidak ada data yang dicetak
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2>
```



### Deskripsi:

Program tersebut merupakan Stack. Program tersebut melakukan operasi seperti push (menambahkan), pop(menghapus), peek(mengubah data pada posisi tertentu), dan destroy (menghapus semua data). Program tersebut menampilkan output hasil dari operasi dengan fungsi cetakArrayBuku. Output yang ditampilkan yaitu hasil dari operasi stack tersebut.

- `#include <iostream>` ➔ merupakan input output stream header yang digunakan sebagai standar input output operasi yang digunakan di c++
- `using namespace std;` ➔ digunakan untuk mendeklarasikan/memberitahukan kepada compiler bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace std
- `string arrayBuku[5];` ➔ mendeklarasikan array string dengan nama array buku dengan ukuran maksimal 5
- `int maksimal = 5, top = 0;` ➔ mendeklarasikan variabel maksimal dengan nilai 5 dan variabel top dengan nilai 0
- `bool isFull()` ➔ fungsi untuk mengecek apakah stack penuh
- `return (top == maksimal);` ➔ mengembalikan true jika top sama dengan maksimal yang berarti stack penuh

- `bool isEmpty()` → fungsi untuk mengecek apakah stack kosong
- `return (top == 0);` → mengembalikan true jika top sama dengan 0 yang berarti stack kosong
- `void pushArrayBuku(string data)` → fungsi untuk menambah data ke stack
- `if (isFull())` → percabangan if jika stack penuh
- `cout << "Data telah penuh" << endl;` → cout menampilkan pernyataan bahwa data telah penuh
- `else` → jika stack belum penuh
- `arrayBuku[top] = data;` → menambahkan data ke posisi top pada stack
- `top++;` → menambah nilai top dengan 1 menaik
- `void popArrayBuku()` → fungsi untuk menghapus data dari stack
- `if (isEmpty())` → percabangan if jika stack kosong
- `cout << "Tidak ada data yang dihapus" << endl;` → cout menampilkan pernyataan bahwa tidak ada data yang dihapus
- `else` → jika stack tidak kosong
- `arrayBuku[top - 1] = "";` → menghapus data pada posisi top - 1 pada stack
- `top--;` → mengurangi nilai tp dengan 1
- `void peekArrayBuku(int posisi)` → fungsi untuk melihat data pada posisi tertentu dalam stack
- `if (isEmpty())` → percabangan if jika stack kosong
- `cout << "Tidak ada data yang bisa dilihat" << endl;` → cout menampilkan tidak ada data yang bisa dilihat
- `else` → jika stack tidak kosong
- `int index = top;` → mendeklarasikan variabel indeks dengan nilai top
- `for (int i = 1; i <= posisi; i++)` → perulangan for dengan i dimulai dari 1; i kurang sama dengan posisi; i++(increment)



- `index--;` → mengurangi nilai indeks dengan 1
- `cout << "Posisi ke " << posisi << " adalah " << arrayBuku[index] << endl;` → cout mencetak data dengan posisi tertentu
- `int countStack()` → fungsi untuk menghitung jumlah data dalam stack
- `return top;` → mengembalikan nilai top
- `void changeArrayBuku(int posisi, string data)` → fungsi untuk mengubah data pada posisi tertentu pada stack
- `if (posisi > top)` → percabangan if jika posisi melebihi jumlah data
- `cout << "Posisi melebihi data yang ada" << endl;` → cout menampilkan statement posisi data yang ada
- `else` → jika posisi tidak melebihi jumlah data dalam stack
- `int index = top;` → mendeklarasikan variabel indeks dengan nilai top
- `for (int i = 1; i <= posisi; i++)` → perulangan for untuk looping sebanyak posisi
- `index--;` → mengurangi nilai indeks dengan 1
- `arrayBuku[index] = data;` → mengubah data pada posisi tertentu
- `void destroyArraybuku()` → fungsi untuk menghapus semua data dalam stack
- `for (int i = top; i >= 0; i--)` → perulangan for dengan i sama dengan top; i lebih dari 0; i—(decrement)
- `arrayBuku[i] = "";` → menghapus data pada posisi i
- `top = 0;` → mengatur nilai top menjadi 0
- `void cetakArrayBuku()` → fungsi untuk mencetak semua data dalam stack
- `if (isEmpty())` → percabangan if jika stack kosong
- `cout << "Tidak ada data yang dicetak" << endl;` → cout menampilkan statement tidak ada data yang dicetak

- else → jika stack tidak kosong
- for (int i = top - 1; i >= 0; i--) → perulangan for dengan i sama dengan top - 1; i lebih dari sama dengan 0; i—(decrement)
- cout << arrayBuku[i] << endl; → cout mencetak data pada posisi i
- int main() → merupakan fungsi utama
- pushArrayBuku("Kalkulus"); → menambahkan kalkulus ke stack
- pushArrayBuku("Struktur Data"); → menambahkan struktur data ke stack
- pushArrayBuku("Matematika Diskrit"); → menambahkan matematika diskrit ke stack
- pushArrayBuku("Dasar Multimedia"); → menambahkan dasar multimedia ke stack
- pushArrayBuku("Inggris"); → menambahkan inggris ke stack
- cetakArrayBuku(); → mencetak semua data dalam stack
- cout << "\n"; → cout mencetak baris baru
- cout << "Apakah data stack penuh? " << isFull() << endl; → cout menampilkan pernyataan Apakah data stack penuh?
- cout << "Apakah data stack kosong? " << isEmpty() << endl; → cout menampilkan pernyataan Apakah data stack kosong?
- peekArrayBuku(2); → melihat data pada posisi 2 dalam stack
- popArrayBuku(); → menghapus data terakhir dalam stack
- cout << "Banyaknya data = " << countStack() << endl; → cout mencetak jumlah data dalam stack
- changeArrayBuku(2, "Bahasa Jerman"); → mengubah data pada posisi 2 menjadi Bahasa jerman
- cetakArrayBuku(); → mencetak semua data dalam stack
- cout << "\n"; → cout mencetak baris baru
- destroyArraybuku(); → menghapus semua data dalam stack

- `cout << "Jumlah data setelah dihapus: " << top << endl;` ➔ `cout` menampilkan jumlah data setelah dihapus
- `cetakArrayBuku();` ➔ mencetak semua data dalam stack
- `return 0;` ➔ program akan mengembalikan (return) nilai 0 ke operating sistem yang menjalankan program tersebut

## B. Unguided/Tugas

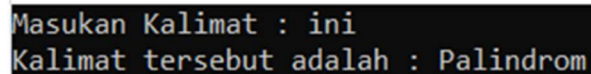
### 1. Unguided 1

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 <stack>
#include <algorithm>

using namespace std;

string reverseString(string sentence)
{
    stack<char> st;
    for (char ch : sentence)
    {
        st.push(ch);
    }

    string reversed_sentence = "";
    while (!st.empty())
    {
        reversed_sentence += st.top();
        st.pop();
    }
    return reversed_sentence;
}
```

```
}

bool isPalindrome (string str)
{
    string reversedStr =
reverseString(str);
    return str == reversedStr;
}

int main()
{
    string sentence;
    cout << "Masukkan kalimat : ";
    getline(cin, sentence);

    sentence.erase(remove(sentence.begin()
, sentence.end(), ' '), sentence.end());
    transform(sentence.begin(),
sentence.end(), sentence.begin(),
::tolower);

    if (isPalindrome(sentence))
    {
        cout << "Kalimat tersebut adalah
palindrom." << endl;
    }
    else
    {
        cout << "Kalimat tersebut bukan
palindrom." << endl;
    }
    return 0;
}
```

Screenshots Output

```
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode
.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MI
Engine-In-nbtxivdn.fv4' '--stdout=Microsoft-MIEngine-Out-zf2hnibh.dew' '--stderr=Microsoft-M
IEngine-Error-autak3e3.44m' '--pid=Microsoft-MIEngine-Pid-gqnvyq3e.nlh' '--dbgExe=c:\mingw32
\bin\gdb.exe' '--interpreter=mi'
Masukkan kalimat : ini
Kalimat tersebut adalah palindrom.
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2>

*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoirddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8
```

```
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode
.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MI
Engine-In-po5yf1u5.z4j' '--stdout=Microsoft-MIEngine-Out-ivxmmka1.ure' '--stderr=Microsoft-M
IEngine-Error-jy4e5nmg.rrm' '--pid=Microsoft-MIEngine-Pid-23ncz0et.v4h' '--dbgExe=c:\mingw32
\bin\gdb.exe' '--interpreter=mi'
Masukkan kalimat : telkom
Kalimat tersebut adalah bukan palindrom.
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2>

*Tidak berjudul - Not...
File Edit Lihat
NIM : 2311102046
Nama : Faisal Khoirddin
Kelas : IF-11-B
Ln 3, Col 16 100% Windows (CRLF) UTF-8
```

### Deskripsi:

Program tersebut program Stack. Program tersebut untuk memeriksa apakah kalimat adalah polindrom atau tidak. Pengguna diminta untuk menginputkan sebuah kalimat kemudian program akan memeriksa inputan kalimat merupakan polindrom atau bukan. Output program tersebut yaitu menampilkan kalimat yang sudah diinputkan merupakan polindrom atau bukan.

### Cara kerja program tersebut

Pertama user menginputkan sebuah kalimat kemudian program tersebut fungsi reverseString untuk membalikkan urutan karakter dalam sebuah string yaitu dengan cara memasukkan setiap karakter dari string ke dalam stack, kemudian mengambil karakter-karakter tersebut dari stack. Dan Menyusun menjadi string baru. Selanjutnya program tersebut menggunakan fungsi isPalindrome yang memeriksa apakah sebuah string adalah

palindrom atau bukan. Dalam fungsi main() program tersebut meminta pengguna memasukkan sebuah kalimat. Kemudian program tersebut memeriksa apakah kalimat tersebut adalah palindrom atau bukan dengan memanggil fungsi isPalindrom. Jika palindrom, program akan mencetak pernyataan “Kalimat tersebut adalah palindrom”. Jika bukan, akan mencetak pernyataan “Kalimat tersebut bukan palindrom”.

- `#include <iostream>` ➔ merupakan input output stream header yang digunakan sebagai standar input output operasi yang digunakan di c++
- `#include <stack>` ➔ digunakan untuk memasukkan library stack
- `#include <algorithm>` ➔ digunakan untuk memasukkan library algorithm
- `using namespace std;` ➔ digunakan untuk mendeklarasikan/memberitahukan kepada compiler bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace std
- `string reverseString(string sentence)` ➔ deklarasi fungsi untuk membalikkan string
- `stack<char> st;` ➔ membuat stack kosong untuk menyimpan karakter
- `for (char ch : sentence)` ➔ perulangan untuk setiap karakter di dalam string
- `st.push(ch);` ➔ memasukkan setiap karakter ke dalam stack
- `string reversed_sentence = "";` ➔ membuat string kosong untuk menyimpan string yang sudah dibalik
- `while (!st.empty())` ➔ perulangan selama stack masih ada isinya
- `reversed_sentence += st.top();` ➔ menambahkan karakter teratas dari stack ke string
- `st.pop();` ➔ menghapus karakter teratas dari stack

- `return reversed_sentence;` ➔ mengembalikan string yang sudah dibalik
- `bool isPalindrome (string str)` ➔ fungsi untuk mengecek apakah string adalah palindrom
- `string reversedStr = reverseString(str);` ➔ membalikkan string
- `return str == reversedStr;` ➔ mengembalikan true jika string asli sama dengan string yang dibalik, dan false jika tidak
- `int main()` ➔ merupakan fungsi utama
- `string sentence;` ➔ membuat string untuk menyimpan kalimat yang akan diperiksa
- `cout << "Masukkan kalimat : ";` ➔ cout menampilkan pernyataan untuk memasukkan kalimat
- `getline(cin, sentence);` ➔ menerima input kalimat dari pengguna
- `sentence.erase(remove(sentence.begin(), sentence.end(), ' '), sentence.end());` ➔ menghapus semua spasi dari kalimat
- `transform(sentence.begin(), sentence.end(), sentence.begin(), ::tolower);` ➔ mengubah semua karakter dalam kalimat menjadi huruf kecil
- `if (isPalindrome(sentence))` ➔ percabangan if jika kalimat adalah palindrom
- `cout << "Kalimat tersebut adalah palindrom." << endl;` ➔ cout menampilkan pernyataan bahwa kalimat adalah palindrom
- `else` ➔ jika bukan kalimat palindrom
- `cout << "Kalimat tersebut bukan palindrom." << endl;` ➔ cout menampilkan pernyataan bahwa kalimat bukan palindrom
- `return 0;` ➔ program akan mengembalikan (return) nilai 0 ke

## 2. 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
```

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

using namespace std;

string reverseString(string sentence)
{
    stack<char> st;
    for (char ch : sentence)
    {
        st.push(ch);
    }

    string reversed_sentence = "";
    while (!st.empty())
    {
        reversed_sentence += st.top();
        st.pop();
    }
    return reversed_sentence;
}

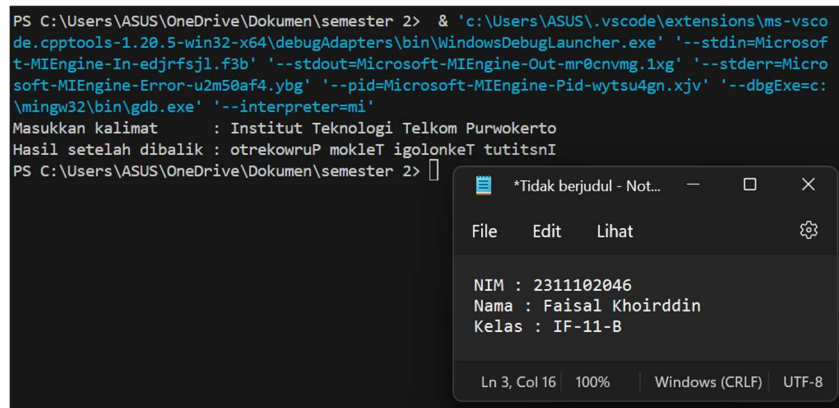
int main()
{
    string sentence;
    cout << "Masukkan kalimat          : ";
    getline(cin, sentence);

    string reversed_sentence =
reverseString(sentence);

    cout << "Hasil setelah dibalik : " <<
reversed_sentence << endl;

    return 0;
}
```

## Screenshot Output



```
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2> & 'c:\Users\ASUS\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-edjrfjsjl.f3b' '--stdout=Microsoft-MIEngine-Out-mr0cnvmg.1xg' '--stderr=Microsoft-MIEngine-Error-u2m50af4.ybg' '--pid=Microsoft-MIEngine-Pid-wytsu4gn.xjv' '--dbgExe=c:\mingw32\bin\gdb.exe' '--interpreter=mi'
Masukkan kalimat      : Institut Teknologi Telkom Purwokerto
Hasil setelah dibalik : otrekowruP mokleT igolonkeT tutitsnI
PS C:\Users\ASUS\OneDrive\Dokumen\semester 2>
```

\*Tidak berjudul - Not...

File Edit Lihat

NIM : 2311102046  
Nama : Faisal Khoirddin  
Kelas : IF-11-B

Ln 3, Col 16 100% Windows (CRLF) UTF-8

Deskripsi:

Deskripsi Ouput :

Program tersebut program Stack. Program tersebut untuk membalikan kalimat dengan menggunakan input dari user contohnya : Institut Teknologi Telkom Purwokerto. Setelah kalimat diinputkan program akan membalikkan kalimat dengan menggunakan fungsi reverseString. Output program tersebut yaitu menampilkan kalimat yang sudah dibalik contohnya : otrekowruP mokleT igolonkeT tutitsnI.

Deskripsi source code :

- `#include <iostream>` ➔ merupakan input output stream header yang digunakan sebagai standar input output operasi yang digunakan di c++
- `using namespace std;` ➔ digunakan untuk mendeklarasikan/memberitahukan kepada compiler bahwa kita akan menggunakan semua fungsi/class/file yang terdapat dalam namespace std
- `string reverseString(string sentence)` ➔ deklarasi fungsi untuk membalikkan string
- `stack<char> st;` ➔ membuta stack kosong untuk menyimpan karakter
- `for (char ch : sentence)` ➔ perulangan untuk setiap karakter di

dalam string

- `st.push(ch);` ➔ memasukkan setiap karakter ke dalam stack
- `string reversed_sentence = "";` ➔ membuat string kosong untuk menyimpan string yang sudah dibalik
- `while (!st.empty())` ➔ perulangan selama stack masih ada isinya
- `reversed_sentence += st.top();` ➔ menambahkan karakter teratas dari stack ke string
- `st.pop();` ➔ menghapus karakter teratas dari stack ke string
- `return reversed_sentence;` ➔ mengembalikan string yang sudah dibalik
- `int main()` ➔ merupakan fungsi utama
- `string sentence;` ➔ membuat string untuk menyimpan kalimat yang akan dibalik
- `cout << "Masukkan kalimat : ";` ➔ cout untuk menginputkan kalimat
- `getline(cin, sentence);` ➔ menerima input kalimat dari pengguna
- `string reversed_sentence = reverseString(sentence);` ➔ memanggil fungsi untuk membalikkan kalimat
- `cout << "Hasil setelah dibalik : " << reversed_sentence << endl;` ➔ mencetak kalimat yang sudah dibalik
- `return 0;` ➔ program akan mengembalikan (return) nilai 0 ke operating sistem yang menjalankan program tersebut

### C. Kesimpulan

Stack adalah struktur data sederhana yang digunakan untuk menyimpan data (mirip dengan Linked Lists). Konsep utamanya adalah LIFO (Last In First Out), benda yang terakhir masuk dalam stack akan menjadi benda pertama yang dikeluarkan dari stack

Beberapa operasi umum pada stack:

- a. Push (Masukkan): Menambahkan elemen ke dalam tumpukan pada posisi paling atas atau ujung.
- b. Pop (Keluarkan): Menghapus elemen dari posisi paling atas atau ujung tumpukan.
- c. Top (Atas): Mendapatkan nilai atau melihat elemen teratas pada tumpukan tanpa menghapusnya.
- d. IsEmpty (Kosong): Memeriksa apakah tumpukan kosong atau tidak. Praktikum Struktur Data dan Algoritma 2
- e. IsFull (Penuh): Memeriksa apakah tumpukan penuh atau tidak (terutama pada implementasi tumpukan dengan kapasitas terbatas). Size (Ukuran): Mengembalikan jumlah elemen yang ada dalam tumpukan.
- f. Peek (Lihat): Melihat nilai atau elemen pada posisi tertentu dalam tumpukan tanpa menghapusnya.
- g. Clear (Hapus Semua): Mengosongkan atau menghapus semua elemen dari tumpukan.
- h. Search (Cari): Mencari keberadaan elemen tertentu dalam tumpukan.

#### **D. Referensi**

[1] Asisten Praktikum, Struktur Data "STACK", WhatsApp, 2024.

[2] Daisma Bali, Mengenal Konsep Stack dalam Struktur Data Beserta Ilustrasi Dan Contoh Pemrogramannya, diakses dari

<https://daismabali.medium.com/mengenal-konsep-stack-dalam-struktur-data-beserta-ilustrasi-dan-contoh-pemrogramannya-e63e444886e9>

[3] Rizki Muliono, S.Kom, M.Kom. Pemrograman C++ Algoritma & Struktur Data. diakses dari

<https://abdulkadir.blog.uma.ac.id/wp-content/uploads/sites/365/2019/02/Modul-Praktikum-Algoritma-Struktur-Data-Merge.pdf>