

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL VI
STACK (TUMPUKAN)**



Disusun Oleh :

NAMA : Aulia Radix Putra Winarko

NIM : 2311102056

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:

- 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).
- 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 1

```
#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;
}

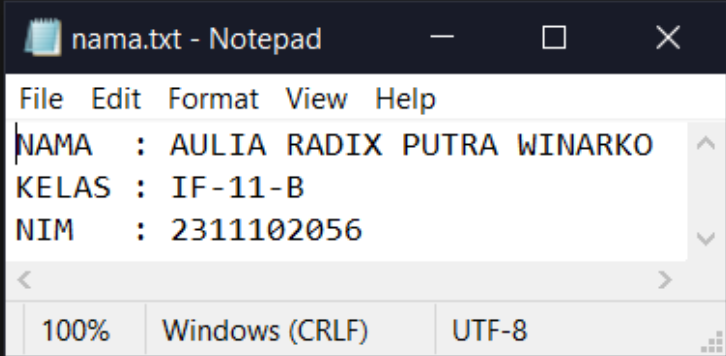
```

Screenshots Output

```
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 D:\C++\Praktikum struktur data\Modul 6>
```



Deskripsi:

Kode di atas menggunakan stack pada proses fungsinya, pertama program akan memasukkan beberapa nama buku dimulai dari kalkulus sampai dasar multimedia, karena stack bersifat LIFO maka kalkulus akan berada dibawah sedangkan dasar multimedia akan berada diatas

C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

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

using namespace std;

string reverseLetters(const string &sentence)
{
    stack<char> letterStack;
    string reversedSentence;
    string word;

    for (char c : sentence)
    {
        if (c == ' ')
        {
            while (!letterStack.empty())
            {
                reversedSentence += letterStack.top();
                letterStack.pop();
            }
            reversedSentence += ' ';
        }
        else
        {
            letterStack.push(c);
        }
    }
    while (!letterStack.empty())
    {
        reversedSentence += letterStack.top();
        letterStack.pop();
    }
    return reversedSentence;
}

bool isPalindrome(const string &sentence)
{

```

```

    stack<char> charStack;
    // Push each character of the word into the stack
    for (char c : sentence)
    {
        charStack.push(c);
    }
    // Compare each character from the stack with the word
    for (char c : sentence)
    {
        if (c != charStack.top())
        {
            return false;
        }
        charStack.pop();
    }
    return true;
}

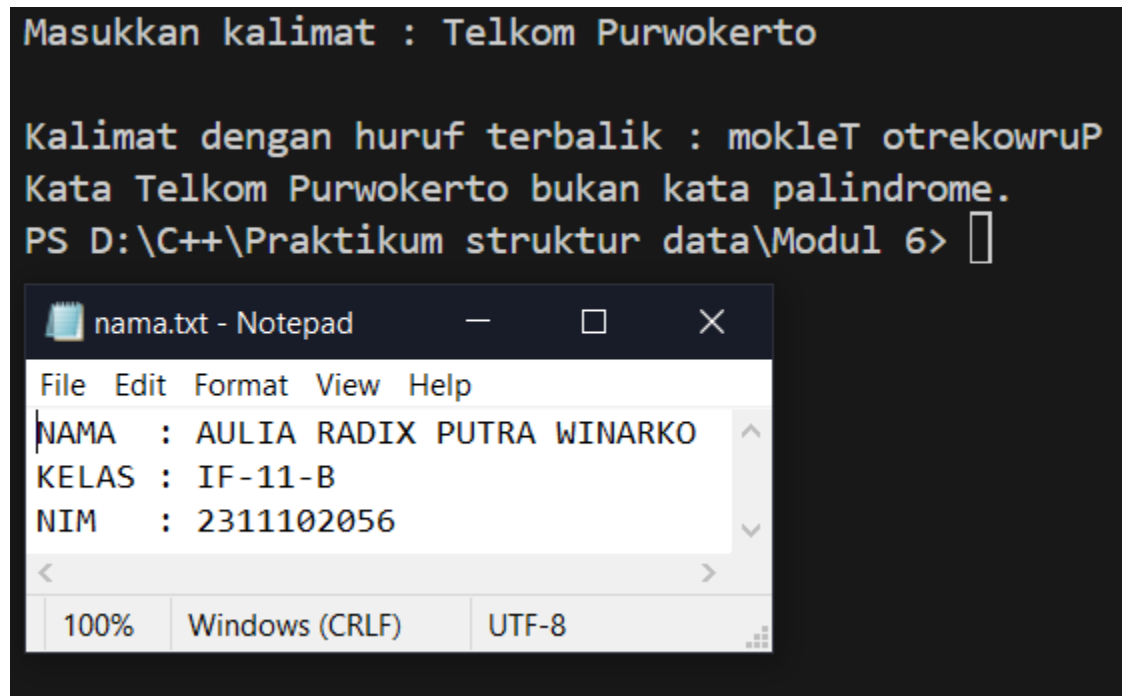
int main()
{
    string sentence;
    cout << "Masukkan kalimat : ";
    getline(cin, sentence);
    cout << endl;
    string reversed = reverseLetters(sentence);
    cout << "Kalimat dengan huruf terbalik : " << reversed << endl;

    if (isPalindrome(sentence))
    {
        cout << "Kata " << sentence << " adalah kata palindrome." <<
endl;
    }
    else
    {
        cout << "Kata " << sentence << " bukan kata palindrome." << endl;
    }

    return 0;
}
}

```


Screenshots Output



The screenshot shows a terminal window with the following output:

```
Masukkan kalimat : Telkom Purwokerto

Kalimat dengan huruf terbalik : mokleT otrekowruP
Kata Telkom Purwokerto bukan kata palindrome.
PS D:\C++\Praktikum struktur data\Modul 6> 
```

Below the terminal window is a Notepad window titled "nama.txt - Notepad". The text inside the Notepad window is:

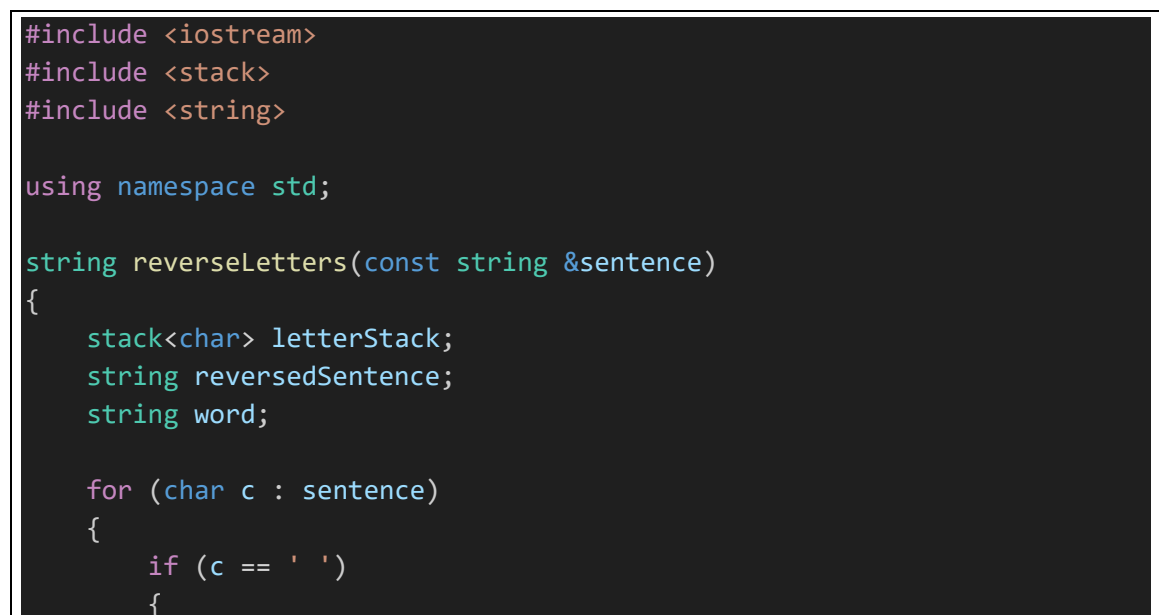
```
File Edit Format View Help
NAMA : AULIA RADIX PUTRA WINARKO
KELAS : IF-11-B
NIM : 2311102056
```

The Notepad window also shows a status bar at the bottom with "100%", "Windows (CRLF)", and "UTF-8".

Deskripsi:

Kode diatas menggunakan stack pada proses fungsinya, pertama program akan memasukkan sebuah kata atau kalimat dan nantinya kata atau kalimat tersebut akan dibalik posisi dan tulisannya melalui proses stack..

Unguided 2



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

using namespace std;

string reverseLetters(const string &sentence)
{
    stack<char> letterStack;
    string reversedSentence;
    string word;

    for (char c : sentence)
    {
        if (c == ' ')
        {
```

```

        while (!letterStack.empty())
        {
            reversedSentence += letterStack.top();
            letterStack.pop();
        }
        reversedSentence += ' ';
    }
    else
    {
        letterStack.push(c);
    }
}
while (!letterStack.empty())
{
    reversedSentence += letterStack.top();
    letterStack.pop();
}
return reversedSentence;
}

bool isPalindrome(const string &sentence)
{
    stack<char> charStack;
    // Push each character of the word into the stack
    for (char c : sentence)
    {
        charStack.push(c);
    }
    // Compare each character from the stack with the word
    for (char c : sentence)
    {
        if (c != charStack.top())
        {
            return false;
        }
        charStack.pop();
    }
    return true;
}

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

```

```

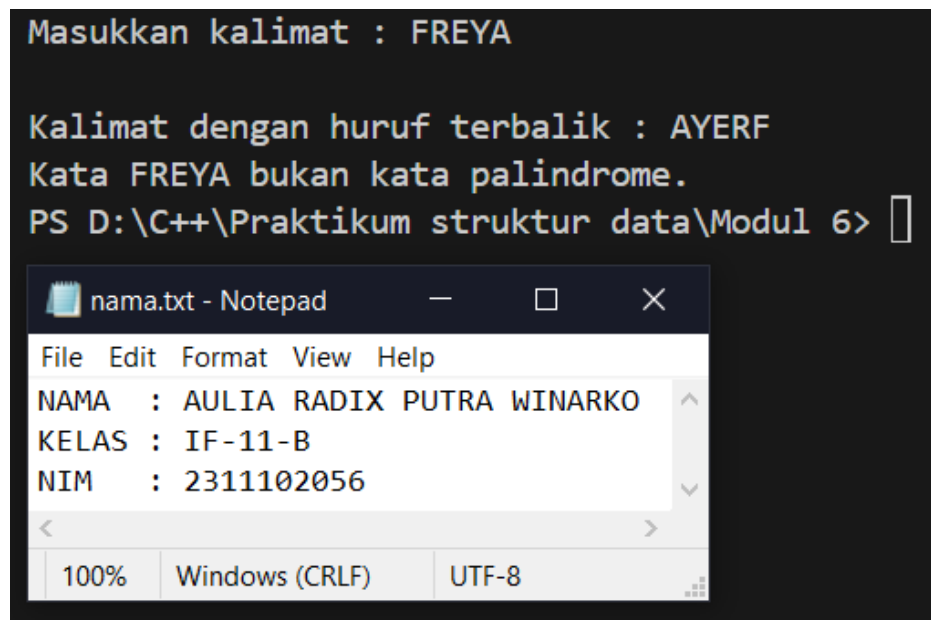
    cout << endl;
    string reversed = reverseLetters(sentence);
    cout << "Kalimat dengan huruf terbalik : " << reversed << endl;

    if (isPalindrome(sentence))
    {
        cout << "Kata " << sentence << " adalah kata palindrome." <<
endl;
    }
    else
    {
        cout << "Kata " << sentence << " bukan kata palindrome." <<
endl;
    }

    return 0;
}

```

Screenshots Output



The screenshot shows the output of a C++ program in a terminal window. The output is as follows:

```

Masukkan kalimat : FREYA

Kalimat dengan huruf terbalik : AYERF
Kata FREYA bukan kata palindrome.
PS D:\C++\Praktikum struktur data\Modul 6>

```

Below the terminal output, there is a Notepad window titled "nama.txt - Notepad". The Notepad window contains the following text:

```

File Edit Format View Help
NAMA : AULIA RADIX PUTRA WINARKO
KELAS : IF-11-B
NIM : 2311102056

```

The Notepad window also shows a status bar at the bottom with "100%", "Windows (CRLF)", and "UTF-8".

Deskripsi:

Kode diatas menggunakan stack pada proses fungsinya, pertama program akan memasukkan sebuah kata atau kalimat dan nantinya kata atau kalimat tersebut akan dicek apakah kata atau kalimat tersebut merupakan Palindrom (kata yang apabila dibalik bunyi atau maknanya sama) atau tidak melalui proses stack.

D. Kesimpulan

Kesimpulan yang dapat diambil dari laporan praktikum dengan “Modul 6 : STACK (TUMPUKAN)” adalah bahwa praktikum ini bertujuan untuk mempelajari fungsi dari penggunaan Stack dalam pemrograman. Praktisi jadi bisa memahami penggunaan dari konsep tersebut dan bagaimana kegunaan dari konsep tersebut dalam program yang dibuat, serta setelah melakukan pratikum, mahasiswa mampu menerapkan konsep tersebut didalam program yang mereka buat. Seperti contohnya praktisi membuat sebuah program untuk menginputkan sebuah kata atau kalimat yang nantinya kata atau kalimat tersebut akan dibalik posisi dengan penyusunan katanya, serta mengecek apakah kata tersebut polindrom atau tidak menggunakan konsep Stack.

Referensi

"Stack Machines: The Emergence of a New Paradigm" by Donald Knuth