

# **LAPORAN PRAKTIKUM**

## **MODUL VI STACK**



**Disusun oleh:  
Brian Farrel Evandhika  
NIM: 2311102037**

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

**PROGRAM STUDI TEKNIK INFORMATIKA  
FAKULTAS INFORMATIKA  
INSTITUT TEKNOLOGI TELKOM PURWOKERTO  
PURWOKERTO  
2023**

# **BAB I**

## **TUJUAN PRAKTIKUM**

### **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

### **B. DASAR TEORI**

Stack adalah struktur data yang terdiri dari serangkaian elemen yang disusun dalam urutan linear, tetapi dengan akses terbatas hanya pada dua operasi utama: penambahan elemen ke puncak stack (push) dan penghapusan elemen dari puncak stack (pop). Konsep ini mirip dengan tumpukan benda di kehidupan sehari-hari di mana Anda hanya dapat menambahkan objek ke atas tumpukan atau mengeluarkan objek dari atasnya. 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. 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.

## 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;
}

```

## Screenshot Output

```
PS C:\Users\MSI GAMING\Documents\Kuliah\SEMESTER 2\PRAKTIKUM STRUKTUR DATA DAN ALGORITME\Praktikum Modul 3\2311102037_Brian-Farrel-Evandhika_IF-11-A_Praktikum-Struktur-Data\Modul 5> cd "c:\Users\MSI GAMING\Documents\Kuliah\SEMESTER 2\PRAKTIKUM STRUKTUR DATA DAN ALGORITME\2311102037_Brian-Farrel-Evandhika_IF-11-A_Praktikum-Struktur-Data\Modul 6\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
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\MSI GAMING\Documents\Kuliah\SEMESTER 2\PRAKTIKUM STRUKTUR DATA DAN ALGORITME\2311102037_Brian-Farrel-Evandhika_IF-11-A_Praktikum-Struktur-Data\Modul 6>
```

## Deskripsi Program

Program ini adalah implementasi struktur data stack menggunakan array dalam bahasa pemrograman C++. Program ini memungkinkan pengguna melakukan operasi dasar pada stack seperti push (menambahkan elemen ke stack), pop (menghapus elemen dari stack), peek (melihat elemen di posisi tertentu), menghitung jumlah elemen dalam stack, mengubah nilai elemen di posisi tertentu, dan menghapus semua elemen dalam stack. Data dalam stack disimpan dalam array, dengan variabel top yang menunjukkan posisi elemen teratas. Program ini juga memiliki fungsi untuk memeriksa apakah stack penuh atau kosong. Pada akhirnya, program ini mencetak isi stack, memeriksa status stack, mengakses elemen pada posisi tertentu, menghapus elemen dari stack, mengubah elemen pada posisi tertentu, dan menghapus semua elemen dari stack sebelum mencetak isi stack yang kosong.

## Unguided

### Unguided 1

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

using namespace std;

bool isPalindrome(string str)
{
    stack<char> s;

    for (char c : str)
    {
        if (isalnum(c))
        {
            s.push(tolower(c));
        }
    }
    for (char c : str)
    {
        if (isalnum(c))
        {
            if (tolower(c) != s.top())
            {
                return false;
            }
            s.pop();
        }
    }

    return true;
}

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

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

```
}  
return 0;  
}
```

### Screenshot Program

```
PS C:\Users\MSI GAMING\Documents\Kuliah\SEMESTER 2\PRAKTIKUM STRUKTUR DATA DAN ALGORITME\Praktikum Modul 3\2311102037_Brian-Farrel-Evandhika_IF-11-A_Praktikum-Struktur-Data\Modul 5> cd "c:\Users\MSI GAMING\Documents\Kuliah\SEMESTER 2\PRAKTIKUM STRUKTUR DATA DAN ALGORITME\2311102037_Brian-Farrel-Evandhika_IF-11-A_Praktikum-Struktur-Data\Modul 6\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }  
Masukkan kalimat: APA  
Kalimat tersebut adalah palindrom.  
PS C:\Users\MSI GAMING\Documents\Kuliah\SEMESTER 2\PRAKTIKUM STRUKTUR DATA DAN ALGORITME\2311102037_Brian-Farrel-Evandhika_IF-11-A_Praktikum-Struktur-Data\Modul 6> █
```

```
PS C:\Users\MSI GAMING\Documents\Kuliah\SEMESTER 2\PRAKTIKUM STRUKTUR DATA DAN ALGORITME\Praktikum Modul 3\2311102037_Brian-Farrel-Evandhika_IF-11-A_Praktikum-Struktur-Data\Modul 5> cd "c:\Users\MSI GAMING\Documents\Kuliah\SEMESTER 2\PRAKTIKUM STRUKTUR DATA DAN ALGORITME\2311102037_Brian-Farrel-Evandhika_IF-11-A_Praktikum-Struktur-Data\Modul 6\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRunnerFile } ; if ($?) { .\tempCodeRunnerFile }  
Masukkan kalimat: Brian  
Kalimat tersebut bukan palindrom.  
PS C:\Users\MSI GAMING\Documents\Kuliah\SEMESTER 2\PRAKTIKUM STRUKTUR DATA DAN ALGORITME\2311102037_Brian-Farrel-Evandhika_IF-11-A_Praktikum-Struktur-Data\Modul 6> █
```

### Deskripsi Program

Program ini adalah sebuah program dalam bahasa C++ yang bertujuan untuk menentukan apakah kalimat yang dimasukkan oleh pengguna merupakan palindrom atau bukan. Palindrom adalah kata, frasa, angka, atau urutan karakter lainnya yang dapat dibaca sama baik dari depan maupun dari belakang. Program ini menggunakan struktur data stack untuk membandingkan setiap karakter dalam kalimat, baik dari depan maupun belakang. Pertama, program membersihkan kalimat dari karakter non-alfanumerik dan mengonversi setiap karakter menjadi huruf kecil. Selanjutnya, program memasukkan setengah pertama dari karakter yang sudah dibersihkan ke dalam stack. Kemudian, program membandingkan setiap karakter dari setengah kedua kalimat dengan karakter yang ada di stack (yang berisi setengah pertama kalimat). Jika setiap karakter cocok (dari kedua sisi kalimat) dan tidak ada karakter tersisa di stack (artinya jumlah karakter di kedua sisi sama), maka program menyatakan bahwa kalimat tersebut adalah palindrom. Jika tidak, program menyatakan bahwa kalimat tersebut bukan palindrom. Program ini memungkinkan pengguna memeriksa apakah kalimat yang dimasukkan adalah palindrom dengan cepat dan efisien menggunakan konsep struktur data stack.



## Unguided 2

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

using namespace std;

string reverseWord(const string &word)
{
    string reversedWord = "";
    for (int i = word.length() - 1; i >= 0; --i)
    {
        reversedWord += word[i];
    }
    return reversedWord;
}

string reverseSentence(const string &sentence)
{
    stack<string> wordStack;
    string word, reversedSentence;
    int start = 0;

    for (int i = 0; i < sentence.length(); ++i)
    {
        if (sentence[i] == ' ')
        {
            word = sentence.substr(start, i - start);
            wordStack.push(reverseWord(word));
            start = i + 1;
        }
    }

    word = sentence.substr(start, sentence.length() - start);
    wordStack.push(reverseWord(word));

    while (!wordStack.empty())
    {
        reversedSentence += wordStack.top() + " ";
        wordStack.pop();
    }

    return reversedSentence;
}

int main()
{
    string sentence;
```

```

    cout << "Masukkan kalimat yang ingin dibalik (minimal 3 kata): ";
    getline(cin, sentence);

    string reversedSentence = reverseSentence(sentence);
    cout << "Hasil Kalimat Terbalik: " << reversedSentence << endl;

    return 0;
}

```

### Screenshot Program

```

PS C:\Users\MSI GAMING\Documents\Kuliah\SEMESTER 2\PRAKTIKUM STRUKTUR DATA DAN ALGORITME\Praktikum M
odul 3\2311102037_Brian-Farrel-Evandhika_IF-11-A_Praktikum-Struktur-Data\Modul 5> cd "c:\Users\MSI G
AMING\Documents\Kuliah\SEMESTER 2\PRAKTIKUM STRUKTUR DATA DAN ALGORITME\2311102037_Brian-Farrel-Evan
dhika_IF-11-A_Praktikum-Struktur-Data\Modul 6\" ; if ($?) { g++ tempCodeRunnerFile.cpp -o tempCodeRu
nnerFile } ; if ($?) { .\tempCodeRunnerFile }
Masukkan kalimat yang ingin dibalik (minimal 3 kata): Brian Farrel Evandhika
Hasil Kalimat Terbalik: akihnavE lerraF nairB
PS C:\Users\MSI GAMING\Documents\Kuliah\SEMESTER 2\PRAKTIKUM STRUKTUR DATA DAN ALGORITME\2311102037_
Brian-Farrel-Evandhika_IF-11-A_Praktikum-Struktur-Data\Modul 6> █

```

### Deskripsi Program

Program tersebut menggunakan struktur data stack dalam bahasa C++ untuk membalikkan urutan kata dalam kalimat yang dimasukkan pengguna. Pada fungsi `reverseWord`, setiap kata dalam kalimat diputar urutannya. Kemudian, fungsi `reverseSentence` membalikkan urutan kata dalam kalimat dengan membagi kalimat menjadi kata-kata, memasukkan setiap kata yang sudah dibalik urutannya ke dalam stack, dan kemudian menggabungkan kembali kata-kata dari stack untuk membentuk kalimat terbalik. Dalam main, kalimat dari pengguna diolah menggunakan `reverseSentence` dan hasilnya ditampilkan. Perubahan program termasuk peningkatan penanganan spasi, kata terakhir dalam kalimat, dan menghilangkan keharusan untuk minimal 3 kata dalam kalimat, sehingga program lebih dapat menangani berbagai masukan kalimat.

## **KESIMPULAN**

Ketiga program tersebut menunjukkan penggunaan struktur data stack dalam bahasa C++ untuk berbagai tujuan. Program pertama merupakan implementasi dasar dari stack menggunakan array, yang memungkinkan pengguna untuk melakukan operasi dasar pada stack seperti push, pop, peek, dan lainnya. Program kedua memanfaatkan stack untuk menentukan apakah sebuah kalimat adalah palindrom, dengan membersihkan dan membandingkan karakter-karakter dalam kalimat. Sementara itu, program ketiga menggunakan stack untuk membalikkan urutan kata dalam sebuah kalimat. Meskipun memiliki tujuan yang berbeda, ketiga program tersebut menunjukkan fleksibilitas dan kegunaan struktur data stack dalam memecahkan masalah yang beragam.

## **Daftar Pustaka**

- [1] <https://www.geeksforgeeks.org/stack-in-cpp-stl/>
- [2] <http://www.nblognlife.com/2014/04/stack-pada-c.html>