

**LAPORAN PRAKTIKUM  
STRUKTUR DATA DAN ALGORITMA**

**MODUL VI  
STACK**



**Disusun Oleh :**  
DIO GILBRAN PRAMANA  
NIM : 2311102062

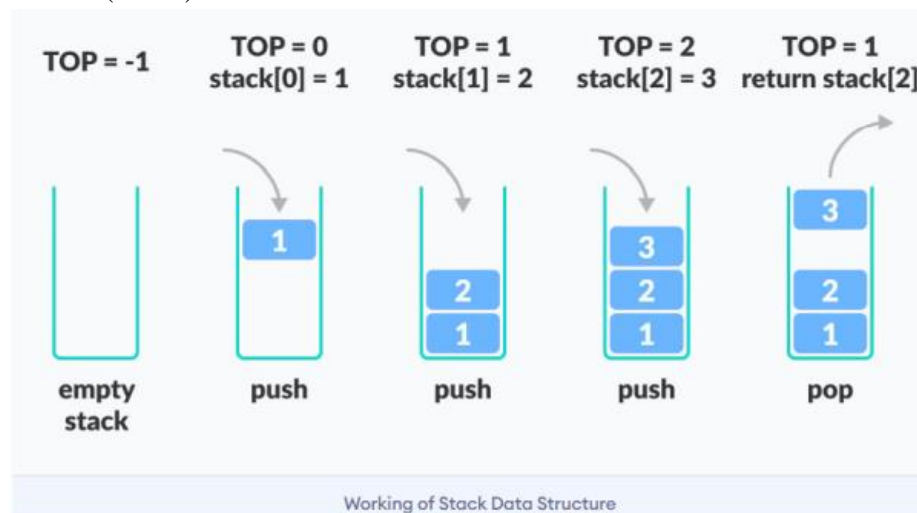
**Dosen**  
WAHYU ANDI SAPUTRA, S.Pd., M.Eng.

**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.
- IsFull (Penuh):** Memeriksa apakah tumpukan penuh atau tidak (terutama pada implementasi tumpukan dengan kapasitas terbatas).
- Size (Ukuran):** Mengembalikan jumlah elemen yang ada dalam tumpukan.
- Peek (Lihat):** Melihat nilai atau elemen pada posisi tertentu dalam tumpukan tanpa menghapusnya.
- Clear (Hapus Semua):** Mengosongkan atau menghapus semua elemen dari tumpukan.
- 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;

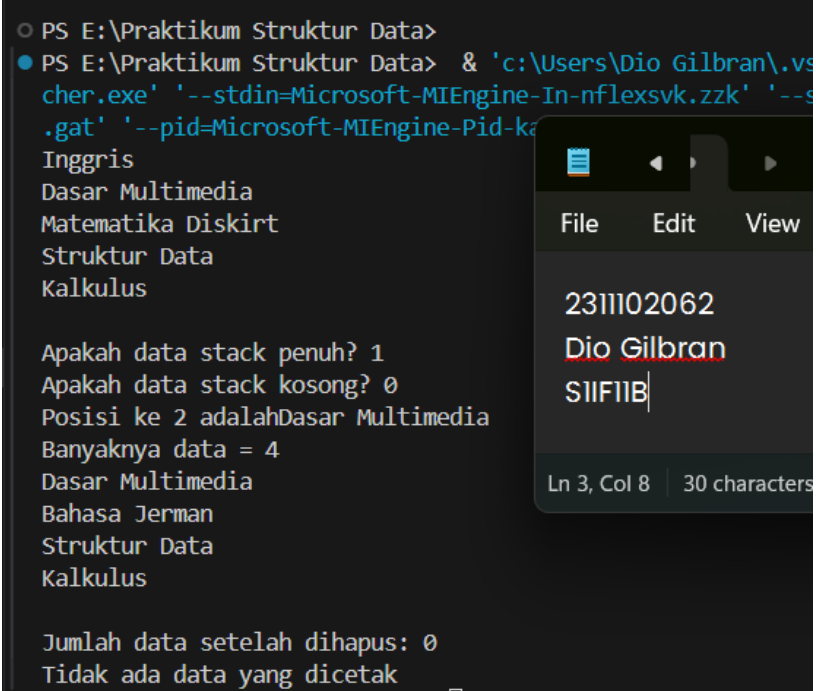
```

```

    }
}
}
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;
}

```

## SCREENSHOT OUTPUT



```

PS E:\Praktikum Struktur Data>
PS E:\Praktikum Struktur Data> & 'c:\Users\Dio Gilbran\.vs
cher.exe' '--stdin=Microsoft-MIEngine-In-nflexsvk.zzk' '--s
.gat' '--pid=Microsoft-MIEngine-Pid-ka
Inggris
Dasar Multimedia
Matematika Diskrit
Struktur Data
Kalkulus

Apakah data stack penuh? 1
Apakah data stack kosong? 0
Posisi ke 2 adalahDasar Multimedia
Banyaknya data = 4
Dasar Multimedia
Bahasa Jerman
Struktur Data
Kalkulus

Jumlah data setelah dihapus: 0
Tidak ada data yang dicetak

```

## DESKRIPSI

Program yang diberikan adalah implementasi dari struktur data stack menggunakan array dalam bahasa pemrograman C++. Stack merupakan struktur data linear yang mengikuti prinsip LIFO (Last In First Out), di mana elemen terakhir yang dimasukkan (push) adalah yang pertama kali dikeluarkan (pop).

## C. UNGUIDED

### UNGUIDED 1

```
#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;
}

```

## SCREENSHOT OUTPUT

```

PS E:\Praktikum Struktur Data> g++ & 'c:\Users\
n-d24honfn.prt' '--stdout=Microsoft-MIEngine
odeBlocks\MinGW\bin\gdb.exe' '--interpreter=
Masukkan kata: roti
Keterangan: kata tersebut bukan palindrom
PS E:\Praktikum Struktur Data>
PS E:\Praktikum Struktur Data> ^C
PS E:\Praktikum Struktur Data>
PS E:\Praktikum Struktur Data> g++ & 'c:\Users\
n-c14yqzwc.udk' '--stdout=Microsoft-MIEngine
odeBlocks\MinGW\bin\gdb.exe' '--interpreter=
Masukkan kata: onono
Keterangan: kata tersebut termasuk palindrom
PS E:\Praktikum Struktur Data>

```

## DESKRIPSI

Program ini memeriksa apakah sebuah kata adalah palindrom dengan menggunakan struktur data stack. Pengguna memasukkan sebuah kata, kemudian program menyimpan setiap karakter dari kata tersebut ke dalam stack. Setelah itu, program membandingkan karakter-karakter dalam kata dengan karakter-karakter yang diambil dari stack (dalam urutan terbalik). Jika semua karakter cocok, maka kata tersebut adalah palindrom dan program akan mencetak bahwa kata tersebut adalah palindrom. Jika tidak, program akan mencetak bahwa kata tersebut bukan palindrom.

## UNGUIDED 2

```
#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;
}

```

#### SCREENSHOT OUTPUT

#### DESKRIPSI

Program ini membalikkan urutan karakter dalam sebuah string menggunakan struktur data stack. Pengguna memasukkan sebuah kata atau kalimat, lalu program menyimpan setiap karakter dari input tersebut ke dalam stack. Setelah itu, program mengambil setiap karakter dari stack (dalam urutan terbalik) untuk membentuk string yang terbalik. Hasilnya, string yang dibalik ditampilkan di layar. Stack memastikan bahwa karakter diambil dalam urutan terbalik dari urutan penyimpanannya, menghasilkan efek pembalikan string.

#### D. KESIMPULAN

Dalam praktikum ini, saya baru saja mengimplementasikan konsep stack melalui dua program berbeda. Program pertama itu memeriksa apakah suatu kata adalah palindrom dengan menyimpan dan memeriksa karakter-karakter dalam stack. Program kedua itu membalikkan urutan karakter dalam sebuah string menggunakan stack. Kedua program tersebut menekankan pentingnya stack sebagai struktur data yang menggunakan prinsip LIFO (Last In, First Out), di mana elemen terakhir yang dimasukkan adalah yang pertama kali dikeluarkan. Dari praktikum ini, saya mendapatkan pemahaman terkait operasi dasar pada stack seperti push, pop, isFull, dan isEmpty, dan lain sebagainya.

#### E. REFERENSI

- Asisten Praktikum, "Modul Stack", GoogleDrive, 2024.
- Karumanchi, N. (2016). Data Structures and algorithms made easy: Concepts, problems, Interview Questions. CareerMonk Publications.
- Budi Raharjo. 2015. Pemrograman C++. Bandung: Informatika.