

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL VI
STACK**



Disusun Oleh :

NAMA : Tegar Serli Arunzika

NIM : 2311102041

Dosen

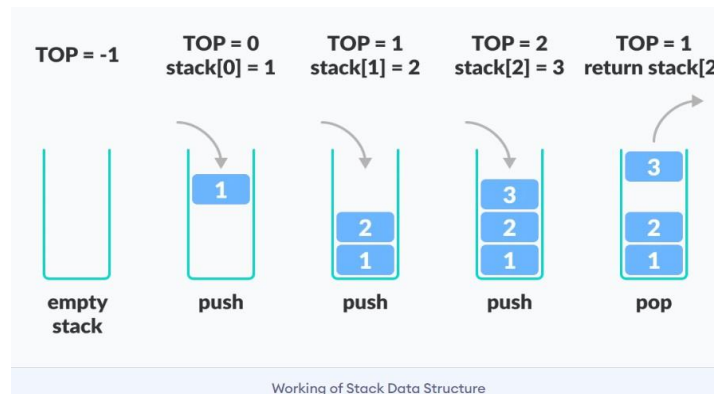
Wahyu Andi Saputra, S.Pd., M.Eng

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFROMATIKA
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 diberikan dan ditempatkan di bagian atas. Ketika sebuah piring dibutuhkan, diambil dari bagian atas tumpukan. Piring yang ditempatkan di tumpukan adalah yang terakhir digunakan.

Definisi: Sebuah tumpukan adalah daftar terurut dimana 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()
{

```

```

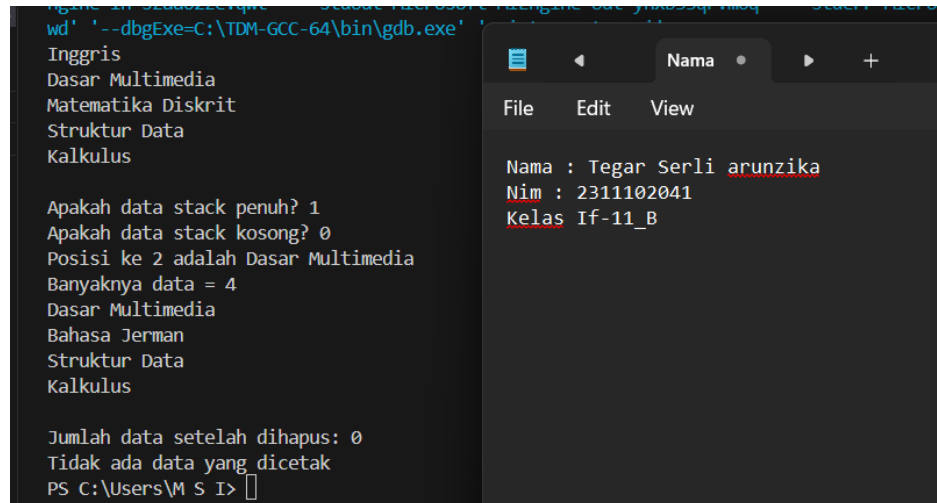
{
    if (isEmpty())
    {
        cout << "Tidak ada data yang dicetak" << endl;
    }
    else
    {
        for (int i = top - 1; i >= 0; i--)
        {
            cout << arrayBuku[i] << endl;
        }
    }
}

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;
}

```

Screenshots Output



The screenshot shows two windows. The left window is a terminal running a C++ program. The output shows a stack of subjects: Inggris, Dasar Multimedia, Matematika Diskrit, Struktur Data, and Kalkulus. It then checks if the stack is full (1) or empty (0), finds the element at index 2 (Dasar Multimedia), and prints the stack contents. The right window is a simple text input form with fields for Nama, Nim, and Kelas, containing the values: Tegar Serli arunzika, 2311102041, and If-11_B.

```
gdb -i /usr/bin/gdb
wd' '--dbgExe=C:\TDM-GCC-64\bin\gdb.exe'
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\M S I>
```

Nama : Tegar Serli arunzika
Nim : 2311102041
Kelas If-11_B

Deskripsi

Di dalam program ini, terdapat fungsi-fungsi dasar untuk operasi pada stack, seperti push (memasukkan elemen ke dalam stack), pop (mengeluarkan elemen dari stack), peek (melihat elemen pada posisi tertentu tanpa menghapusnya), dan beberapa fungsi lainnya.

Berikut adalah deskripsi singkat dari beberapa fungsi dan bagian penting dari program:

1. `pushArrayBuku(string data)`: Fungsi ini digunakan untuk memasukkan data ke dalam stack. Jika stack penuh, akan ditampilkan pesan "Data telah penuh".
2. `popArrayBuku()`: Fungsi ini menghapus elemen teratas dari stack. Jika stack kosong, akan ditampilkan pesan "Tidak ada data yang dihapus".
3. `peekArrayBuku(int posisi)`: Fungsi ini digunakan untuk melihat elemen pada posisi tertentu dalam stack tanpa menghapusnya. Jika stack kosong, akan ditampilkan pesan "Tidak ada data yang bisa dilihat".
4. `changeArrayBuku(int posisi, string data)`: Fungsi ini mengganti nilai elemen pada posisi tertentu dalam stack dengan nilai baru. Jika posisi yang dimasukkan melebihi jumlah data yang ada dalam stack, akan ditampilkan pesan "Posisi melebihi data yang ada".
5. `destroyArraybuku()`: Fungsi ini menghapus semua elemen dalam stack dan mengembalikan stack ke kondisi awal.
6. `cetakArrayBuku()`: Fungsi ini mencetak semua elemen dalam stack dari atas ke bawah.

Di dalam fungsi `main()`, beberapa operasi dasar pada stack dilakukan seperti memasukkan data ke dalam stack, menghapus data dari stack, mengubah data dalam stack, mencetak isi stack, dan menghapus semua data dari stack.

C. Unguided/Tugas

Unguided 1

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

using namespace std;

class Stack {
public:
    Stack() : top(nullptr), size(0) {}

    void push(char a) {
        Node* tmp = new Node;
        tmp->data = a;
        tmp->next = top;
        top = tmp;
        size++;
    }

    char pop() {
        if (size == 0) {
            cerr << "Stack is empty!" << endl;
            exit(1);
        }
        char data = top->data;
        Node* tmp = top;
        top = top->next;
        delete tmp;
        size--;
        return data;
    }

    bool isEmpty() const {
        return size == 0;
    }

private:
    struct Node {
        char data;
        Node* next;
    };
};
```

```

Node* top;
    int size;
};

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

    for (char c : input) {
        stack.push(c);
    }

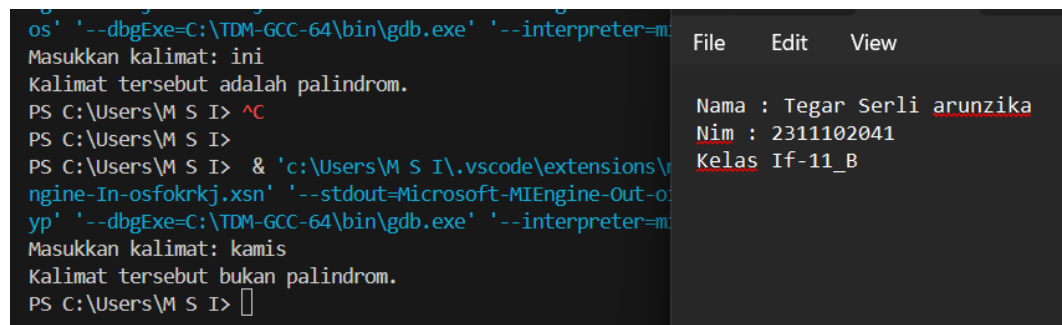
    string reversed;
    while (!stack.isEmpty()) {
        reversed += stack.pop();
    }

    bool isPalindrome = (input == reversed);
    if (isPalindrome) {
        cout << "Kalimat tersebut adalah palindrom." << endl;
    } else {
        cout << "Kalimat tersebut bukan palindrom." << endl;
    }

    return 0;
}

```


Screenshots Output



```
os' '--dbgExe=C:\TDM-GCC-64\bin\gdb.exe' '--interpreter=m
Masukkan kalimat: ini
Kalimat tersebut adalah palindrom.
PS C:\Users\M S I> ^C
PS C:\Users\M S I>
PS C:\Users\M S I> & 'c:\Users\M S I\.vscode\extensions\
ngine-In-osfokrkj.xsn' '--stdout=Microsoft-MIEngine-Out-o
yp' '--dbgExe=C:\TDM-GCC-64\bin\gdb.exe' '--interpreter=m
Masukkan kalimat: Kamis
Kalimat tersebut bukan palindrom.
PS C:\Users\M S I> 
```

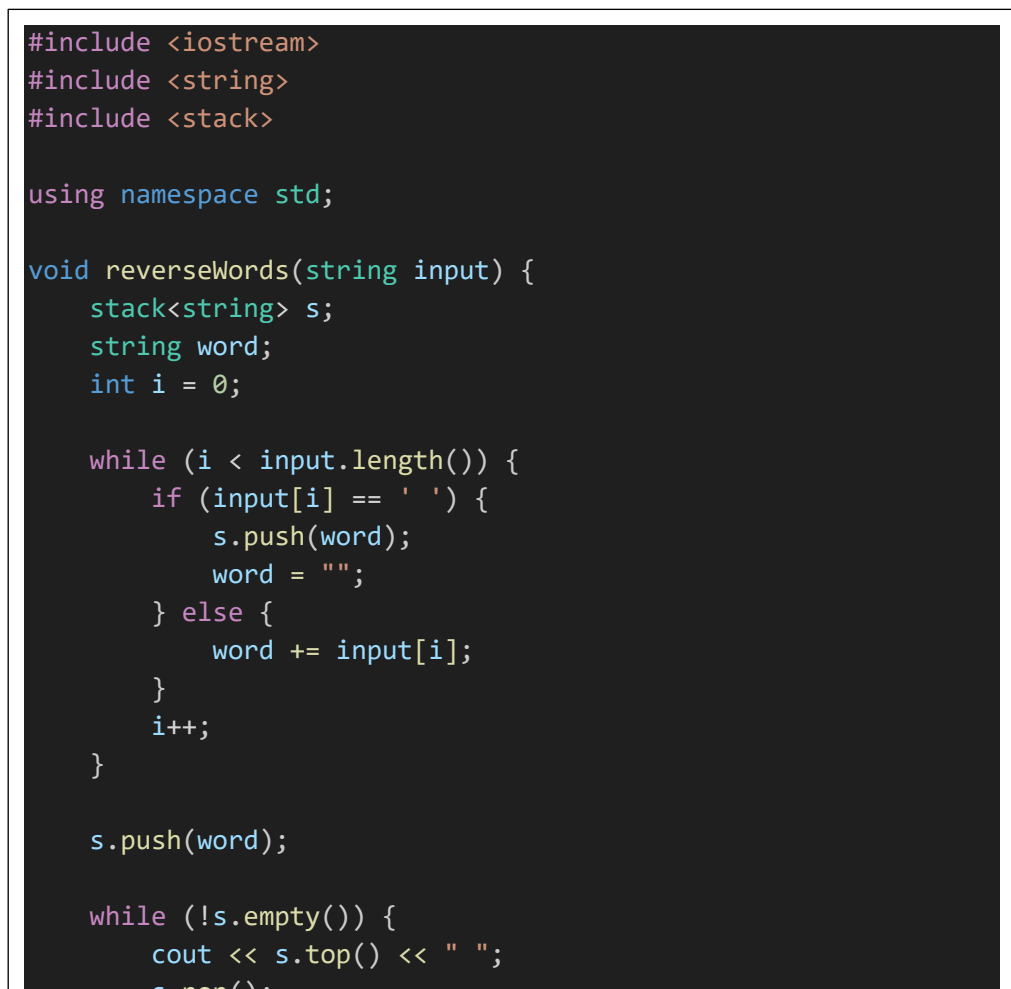
```
File Edit View

Nama : Tegar Serli arunzika
Nim : 2311102041
Kelas If-11_B
```

Deskripsi

Program ini adalah sebuah implementasi dalam bahasa C++ untuk menentukan apakah sebuah kalimat adalah palindrom atau tidak menggunakan struktur data stack. Program ini bekerja dengan cara memasukkan setiap karakter dari kalimat ke dalam stack, kemudian mengeluarkan karakter-karakter tersebut satu per satu untuk membuat kalimat yang terbalik. Setelah itu, program membandingkan kalimat asli dengan kalimat yang sudah dibalik. Jika kedua kalimat tersebut sama, maka kalimat tersebut adalah palindrom. Jika tidak, maka bukan palindrom.

Uguided 2



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

using namespace std;

void reverseWords(string input) {
    stack<string> s;
    string word;
    int i = 0;

    while (i < input.length()) {
        if (input[i] == ' ') {
            s.push(word);
            word = "";
        } else {
            word += input[i];
        }
        i++;
    }

    s.push(word);

    while (!s.empty()) {
        cout << s.top() << " ";
        s.pop();
    }
}
```

```

        s.pop();
    }
}

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

    cout << "Original: " << input << endl;
    cout << "Reversed: ";
    reverseWords(input);
    cout << endl;

    return 0;
}

```

Screenshots Output

```

gdb
PS C:\Users\M S I>
Masukkan kalimat: Cilacap Bercahaya cuy
Original: Cilacap Bercahaya cuy
Reversed: cuy Bercahaya Cilacap
PS C:\Users\M S I>

```

Kesimpulan

Program ini merupakan sebuah program dalam bahasa C++ yang mengimplementasikan fungsi untuk membalikkan urutan kata dalam sebuah kalimat. Program ini menggunakan struktur data stack untuk menyimpan setiap kata dalam kalimat, kemudian mengeluarkannya dari stack satu per satu untuk mencetak kalimat yang sudah dibalik.

D. Referensi

Asisten Praktikum. (2024). Modul VI: Stack

Karumanchi, N. (2016). Data Structures and algorithms made easy: Concepts, problems, Interview Questions. CareerMonk Publications.