

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL VI
STACK**



Disusun Oleh :

NAMA : Ahmad Titana Nanda Pramudya

NIM : 2311102042

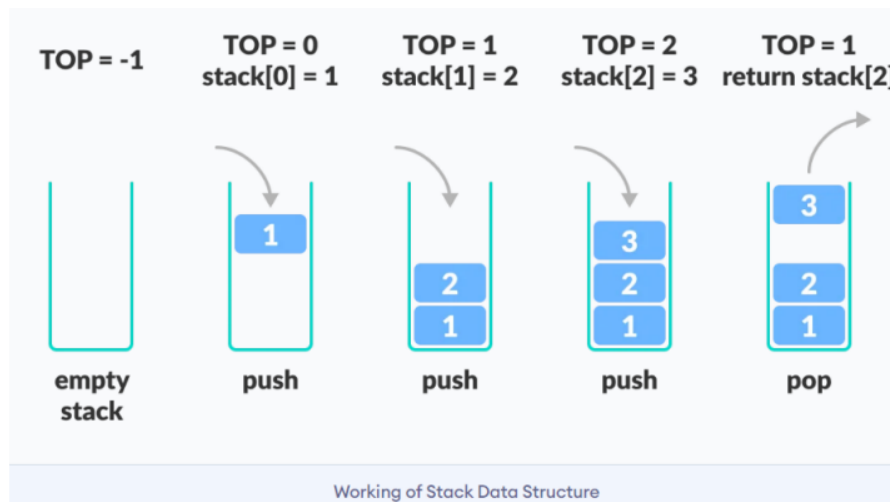
Dosen

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 teratur 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.

Stack C++ – Stack yang berarti tumpukan, dalam bahasa pemrograman merupakan sebuah tumpukan atau kumpulan data dimana data yang diletakkan di atas data yang lain. Stack menggunakan konsep yang dinamakan LIFO Last In First Out. atau bias dengan istilah, element terakhir yang masuk dalam antrian, akan menjadi element pertama yang akan diambil atau diseleksi. Dalam proses komputerisasi, ada namanya operasi Push dan operasi pop. untuk meletakkan Sebuah elemen pada bagian atas dari stack, maka dilakukan operasi push. Dan untuk memindahkan dari tempat yang atas tersebut, maka dilakukan operasi pop. Disini kita akan membahas sedikit mengenai stack serta contoh program stack C++. Stack adalah semua akses dibatasi pada elemen yang paling akhir di sisipkan (Fasikom UI, 2010). Stack adalah sebuah kumpulan data dimana data yang diletakkan diatas data yang lain (Kadir,2013). Kesimpulannya adalah bahwa stack itu merupakan tumpukan data dimana data tersebut saling bertumpuk antar satu dengan yang lainnya yang mana stack ini memiliki sifat LIFO (Last In First Out) yang mana maksud dari sifat ini adalah diman data yang terakhir masuk dan data tersebut yang pertama kali keluar Stack sendiri memiliki istilah-istilah yang sangat penting didalam stack tersebut. Berikut merupakan istilah-istilah yang terdapat dalam stack:

1. Create

Operasi untuk membuat sebuah stack kosong.

```
struct Stack {  
    int top;  
    int data[n];  
  
} myStack;
```

2. InitStack

Operasi untuk menginisialisasikan nilai awal stack (Stack Kosong).

```
void initStack(){  
    myStack.top = -1  
  
}
```

3. isEmpty

Operasi untuk menginisialisasikan nilai awal stack (stack kosong). Stack kosong ditandai dengan nilai Top kurang dari nol (-1).

```
int isEmpty()  
{  
    if (myStack.top == -1)  
        return 1; //true  
    else  
        return 0; //false  
}
```

4. isFull

Operasi untuk memeriksa apakah stack yang ada sudah penuh. Stack akan mengindikasikan penuh jika puncak stack (top) terletak tepat di bawah jumlah maksimum (MAX) yang dapat ditampung stack.

```
int isFull()
{
    if (myStack.top == MAX-1)
        return 1; //true
    else
        return 0; //false
}
```

5. Push (Insert)

Operasi untuk menambahkan satu elemen ke dalam stack dan tidak dapat dilakukan jika stack dalam keadaan penuh.

```
void push (int data
{
    if (isEmpty() == 1){
        myStack.top++;
        myStack.data[myStack.top] = Data;
    }
    else if (isFull() == 0){
        myStack.top++;
        myStack.data[myStack.top] = Data;
    }
    else{
        cout << "Stack sudah penuh!";
    }
}
```

6. Pop

Operasi untuk mengambil atau menghapus data teratas (top) dari stack.

```
void pop() {
    if (isEmpty() == 0){
        myStack.top--;
        cout << " Data teratas terambil" << endl;
    } else{
        cout << " Stack masih kosong!" << endl;
    }
}
```

7. Clear

Operasi untuk menghapus atau mengosongkan seluruh data stack. Jika Top bernilai -1 maka stack dianggap kosong.

```
void clear(){
    myStack.top = -1;
}
```

8. Display

Operasi untuk menampilkan seluruh data stack.

```
void display() {
    if (isEmpty() == 0){
        for (int i= myStack.top; i>=0; i--){
            cout << myStack.data[i] << endl;
        }
    }
    else{
        cout << "Stack masih kosong!" << endl;
    }
}
```

B. Guided

Sour 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 :

```

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:\contoh>

```


Deskripsi:

Program di atas adalah implementasi stack menggunakan array dalam bahasa pemrograman C++. Stack adalah struktur data yang mengikuti prinsip LIFO (Last In, First Out). Program ini terdiri dari beberapa fungsi untuk mengelola stack, seperti menambah (push), menghapus (pop), melihat data tertentu (peek), mengganti data, menghitung elemen dalam stack, menghancurkan seluruh data dalam stack, dan mencetak elemen-elemen dalam stack.

C. Unguided

Unguided 1

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

using namespace std;

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

    for (char ch : str) {
        if (isalpha(ch)) {
            cleanedStr += tolower(ch);
        }
    }

    for (char ch : cleanedStr) {
        s.push(ch);
    }

    for (char ch : cleanedStr) {
        if (ch != s.top()) {
            return false;
        }
        s.pop();
    }

    return true;
}
```

```

int main() {
    string input;

    cout << " kalimat: ";
    getline(cin, input);

    if (isPalindrome(input)) {
        cout << "Kalimat \"" << input << "\" adalah : palindrom" <<
endl;
    } else {
        cout << "Kalimat \"" << input << "\" bukan palindrom" << endl;
    }

    return 0;
}

```

Screenshot Output:

<pre> gExe=C:\msys64\ucrt64\bin\gdb.exe kalimat: Telkom Kalimat "Telkom" bukan palindrom PS D:\contoh> </pre>	<pre> Nama : Ahmad Titana Nanda Pramudya Nim : 2311102042 Kelas: IF-11-B </pre>
<pre> gExe=C:\msys64\ucrt64\bin\gdb.exe kalimat: ini Kalimat "ini" adalah : palindrom PS D:\contoh> </pre>	<pre> Nama : Ahmad Titana Nanda Pramudya Nim : 2311102042 Kelas: IF-11-B </pre>

Deskripsi:

Penjelasan Cara Kerja Program:

1. Membersihkan Kalimat:

- Program mengambil input dari pengguna dan menyimpan dalam variabel input.
- Program membersihkan kalimat dengan menghapus spasi, tanda baca, dan mengonversi semua huruf menjadi huruf kecil. Ini dilakukan dalam loop pertama di dalam fungsi isPalindrome.

2. Mengisi Stack:

- Setelah kalimat dibersihkan, setiap karakter dimasukkan ke dalam stack. Stack adalah struktur data LIFO (Last In First Out), sehingga karakter pertama yang dimasukkan akan berada di bagian bawah stack dan karakter terakhir yang dimasukkan akan berada di

bagian atas.

3. Membandingkan Karakter:

- Program kemudian membandingkan karakter satu per satu dari awal kalimat yang dibersihkan dengan karakter yang diambil dari stack (karakter terakhir yang dimasukkan).
- Jika ada karakter yang tidak cocok, program mengembalikan false, menunjukkan bahwa kalimat bukan palindrom.
- Jika semua karakter cocok, program mengembalikan true, menunjukkan bahwa kalimat adalah palindrom.

4. Menampilkan Hasil:

- Di dalam fungsi main, berdasarkan hasil dari fungsi isPalindrome, program menampilkan apakah kalimat yang diinput adalah palindrom atau bukan.

Unguided 2

Sourcode :

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

using namespace std;

string reverseWords(const string &str) {
    stack<char> s;
    string result = "";

    // Membalikkan setiap karakter dalam string menggunakan stack
    for (char ch : str) {
        s.push(ch);
    }

    // Mengambil karakter dari stack untuk membentuk kalimat yang
    dibalik
    while (!s.empty()) {
        result += s.top();
        s.pop();
    }

    return result;
}
```

```

int main() {
    string input;

    cout << "Masukkan kalimat (minimal 3 kata): ";
    getline(cin, input);

    // Memastikan bahwa input memiliki minimal 3 kata
    int wordCount = 0;
    for (char ch : input) {
        if (ch == ' ') {
            wordCount++;
        }
    }
    wordCount++; // Menambah satu untuk kata terakhir

    if (wordCount < 3) {
        cout << "Error: Kalimat harus memiliki minimal 3 kata." << endl;
    } else {
        string reversed = reverseWords(input);
        cout << "Hasil: " << reversed << endl;
    }

    return 0;
}

```

Screenshot Output :

The screenshot shows a Windows command prompt window with the following text:

```

gExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Masukkan kalimat (minimal 3 kata): Institut Telkom Purwokerto
Hasil: otrekowruP mokleT tutitsnI
PS D:\contoh>

```

Overlaid on the bottom right is a Notepad window titled "Nama Ah". The text inside the Notepad window is:

```

Nama : Ahmad Titana Nanda Pramudya
Nim  : 2311102042
Kelas: IF-11-B

```

Deskripsi :

Program ini bertujuan untuk membalikkan urutan karakter dalam sebuah kalimat yang diinputkan oleh pengguna, dengan syarat kalimat tersebut harus memiliki minimal tiga kata.

Program menggunakan stack untuk membalikkan urutan karakter dalam kalimat

- Stack: Digunakan untuk membalikkan urutan karakter dalam string.

- getline: Digunakan untuk mengambil input kalimat dari pengguna.

- Push: Menambahkan karakter ke dalam stack.

- Pop: Menghapus karakter dari stack dan menambahkannya ke string hasil.

- top: Mengambil karakter teratas dari stack tanpa menghapusnya.

- isalpha & tolower: Digunakan dalam contoh program sebelumnya untuk membersihkan string, tapi tidak digunakan dalam program ini.

- empty(): Memeriksa apakah stack kosong.

D. Kesimpulan

Stack merupakan sebuah kumpulan data yang diletakkan di atas data lainnya, seperti sebuah tumpukan. Dengan demikian, stack merupakan salah satu struktur data yang menerapkan prinsip LIFO (Last In First Out). Dimana elemen yang terakhir disimpan dalam stack, menjadi elemen yang pertama diambil. Ada beberapa operasi dasar pada stack yaitu : Create, initStack, isEmpty, isFull, Push, Pop, clear dan display. Program Stack dapat dipadukan dengan Structure, array dan Single Linked list.

E. Referensi

1. <https://republiccode.com/contoh-program-stack-c/> (STACK C++, Cahaya aprillia (2021)).
2. <https://elektro.um.ac.id/wp-content/uploads/2016/04/ASD-Modul-4-STACK.pdf> (Departemen Teknik Elektro Dan Informatika UM)