

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL VI
STACK**



Disusun Oleh:

Arvan Murbiyanto

2311102074

Dosen

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

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

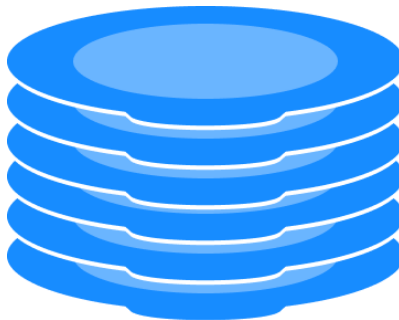
Modul VI

A. Tujuan

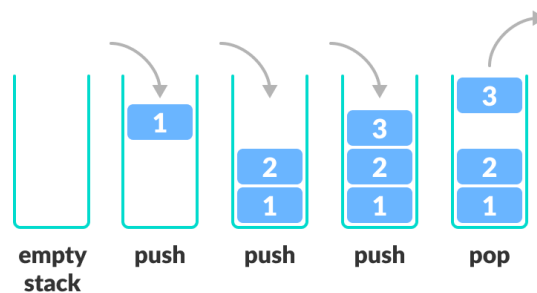
1. Mahasiswa dapat memahami konsep **STACK**
2. Mahasiswa mampu mengimplementasikan **STACK**

B. Dasar Teori

Stack atau dalam Bahasa Indonesia diartikan tumpukan, adalah struktur data linier yang mengikuti prinsip **Last In First Out (LIFO)**. Artinya elemen yang terakhir disisipkan akan menjadi elemen pertama yang keluar. Cara struktur data stack dalam menyimpan sebuah nilai dapat kita bayangkan seperti piring yang disusun rapi secara bertumpuk ke atas. Apabila kita ingin mengambil piring bagian bawah, kita harus terlebih dahulu menyisihkan semua piring yang ada di atas.



Dalam istilah pemrograman, upaya menambahkan elemen pada struktur data stack disebut dengan **push**. Sedangkan proses menghapus atau menghilangkan elemen data dari stack disebut **pop**.



Dari gambar di atas, dapat terlihat bahwa meskipun elemen ke-3 adalah yang paling terakhir ditambahkan, namun elemen tersebut justru yang pertama dihapus. Operasi inilah yang kemudian disebut sebagai prinsip operasi LIFO (Last In First Out). Perlu diketahui, dalam struktur data stack ada dua kondisi yang perlu dihindari, yaitu **underflow** dan **overflow**.

- **Stack underflow**, yaitu keadaan dimana kita mencoba mengakses atau menghapus elemen data pada stack yang kosong
- **Stack overflow**, yaitu keadaan di mana ruang memori yang dialokasikan untuk struktur data stack sudah penuh namun masih dilakukan operasi penyisipan elemen

Jenis-jenis Stack

Berdasarkan kemampuan menyimpan data, struktur data stack dapat dibagi menjadi 2 jenis, yaitu: **register stack** dan **memory stack**.

1. Register stack

Register stack merupakan stack yang hanya mampu menampung data dalam jumlah yang kecil. Kedalaman maksimum pada register stack cenderung dibatasi karena ukuran unit memorinya sangat kecil dibandingkan dengan memory stack.

2. Memory stack

Pada stack jenis ini, kedalaman dari stack cukup fleksibel dan mampu menangani dalam skala yang lebih besar dibandingkan jenis sebelumnya.

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";
}

```

Output:

```

; AT (pf) { .\gugueupero }
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

PS C:\Users\ASUS\OneDrive\Desktop\praktikum struktur data\.vscode>

```

Deskripsi:

Program di atas merupakan implementasi sederhana dari stack dalam bahasa C++. Yang mendefinisikan array arrayBuku dengan kapasitas 5 dan menginisialisasi variabel top dengan nilai 0. Fungsi-fungsi seperti pushArrayBuku, popArrayBuku, peekArrayBuku, dan lainnya digunakan untuk mengelola data dalam stack. Di dalam main(), beberapa data ditambahkan ke stack dan operasi lainnya.

C. UNGUIDED

1. Buatlah program untuk menentukan apakah kalimat tersebut yang diinputkan dalam program stack adalah palindrom/tidak. Palindrom kalimat yang dibaca dari depan dan belakang sama. Jelaskan bagaimana cara kerja programnya.

SOURCE CODE

```
#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;
```



```

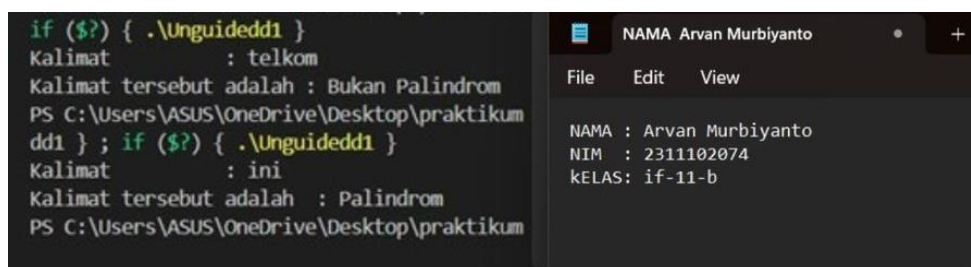
    cout << "Kalimat\t\t : ";
    getline(cin, input);

    if (isPalindrome(input)) {
        cout << "Kalimat tersebut adalah\t : Palindrom" << endl;
    } else {
        cout << "Kalimat tersebut adalah : Bukan Palindrom" <<
endl;
    }

    return 0;
}

```

SCREENSHOT OUTPUT



The screenshot shows two windows. On the left is a terminal window with the following output:

```

if ($?) { .\Unguidedd1 }
Kalimat      : telkom
Kalimat tersebut adalah : Bukan Palindrom
PS C:\Users\ASUS\OneDrive\Desktop\praktikum
dd1 } ; if ($?) { .\Unguidedd1 }
Kalimat      : ini
Kalimat tersebut adalah : Palindrom
PS C:\Users\ASUS\OneDrive\Desktop\praktikum

```

On the right is a Notepad window titled "NAMA Arvan Murbiyanto" with the following text:

```

File Edit View

NAMA : Arvan Murbiyanto
NIM : 2311102074
KELAS: if-11-b

```

DESKRIPSI:

Pada Kode di atas mengimpor beberapa pustaka standar, termasuk `<iostream>` dan `<stack>`, Fungsi pada `isPalindrome` memeriksa apakah string yang diberikan merupakan palindrom atau bukan, String input diubah menjadi huruf kecil dan karakter non-alfabet diabaikan, Karakter-karakter dari string yang telah dibersihkan dimasukkan ke dalam tumpukan (stack), Kemudian, karakter-karakter tersebut dibandingkan dengan karakter teratas dari tumpukan. Jika ada perbedaan, string bukan palindrom dan Jika semua karakter cocok, string dianggap palindrom maka hasil akhir dicetak ke layar.

2. Buatlah program untuk melakukan pembalikan terhadap kalimat menggunakan stack dengan minimal 3 kata. Jelaskan output program dan source codenya beserta operasi/fungsi yang dibuat?

SOURCE CODE

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

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

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

    while (!s.empty()) {
        result += s.top();
        s.pop();
    }

    return result;
}

int main() {
    string input;

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

    int wordCount = 0;
    for (char ch : input) {
        if (ch == ' ') {
            wordCount++;
        }
    }
    wordCount++;
    if (wordCount < 3) {
        cout << "Error: Kalimat harus memiliki minimal 3 kata." <<
endl;
```

```

    } else {
        string reversed = reverseWords(input);
        cout << "Hasil: " << reversed << endl;
    }
    return 0;
}

```

SCRENSHOOT OUTPUT

```

PS C:\Users\ASUS\OneDrive\Desktop\praktikum struk
dd2 } ; if ($?) { .\Unguidedd2 }
Masukkan kalimat : ini telkom bandung
Hasil: gnudnab moklet ini
PS C:\Users\ASUS\OneDrive\Desktop\praktikum struk
dd2 } ; if ($?) { .\Unguidedd2 }
Masukkan kalimat : ini be
Error: Kalimat harus memiliki minimal 3 kata.
PS C:\Users\ASUS\OneDrive\Desktop\praktikum struk

```

NAMA Arvan Murbiyanto

File Edit View

NAMA : Arvan Murbiyanto
NIM : 2311102074
KELAS: if-11-b

DESKRIPSI :

Fungsi `isPalindrome` memeriksa apakah string yang diberikan merupakan palindrom (dapat dibaca sama dari depan dan belakang). String input diubah menjadi huruf kecil dan hanya karakter alfabet yang diperhitungkan. Karakter-karakter dimasukkan ke dalam stack untuk membandingkan apakah palindrom atau bukan.

D. KESIMPULAN

Stack atau dalam Bahasa Indonesia diartikan tumpukan, adalah struktur data linier yang mengikuti prinsip Last In First Out (LIFO), Operasi inilah yang kemudian disebut sebagai prinsip operasi LIFO (Last In First Out). Perlu diketahui, dalam struktur data stack ada dua kondisi yang perlu dihindari, yaitu underflow dan overflow

QUEUE :

Queue adalah struktur data yang mengikuti prinsip FIFO (First In, First Out).

- Elemen pertama yang dimasukkan ke dalam queue akan menjadi elemen pertama yang dikeluarkan.
- Operasi dasar pada queue meliputi enqueue (menambahkan elemen di belakang queue) dan dequeue (mengeluarkan elemen dari depan queue).
- Contoh penggunaan queue adalah dalam antrian proses (seperti antrian printer) dan algoritma pencarian (seperti BFS).

STACK :

- Stack adalah struktur data yang mengikuti prinsip LIFO (Last In, First Out).
- Elemen terakhir yang dimasukkan ke dalam stack akan menjadi elemen pertama yang dikeluarkan.
- Operasi dasar pada stack meliputi push (menambahkan elemen ke atas stack) dan pop (mengeluarkan elemen dari atas stack).
- Contoh penggunaan stack adalah dalam evaluasi ekspresi matematika (menggunakan postfix notation) dan manajemen memori dalam pemrograman.

E. REFERENSI

[1] Asprak “Modul 6 Stack”. Learning Meaning System

[2] trivusi.web.id(2022, 07) struktur data stack

<https://www.trivusi.web.id/2022/07/struktur-data-stack.html>