

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL VI
PENGENALAN STACK**



Disusun Oleh :

NAMA : Pandia Arya Brata

NIM : 2311102076

Dosen

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

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

MODUL 6

STACK

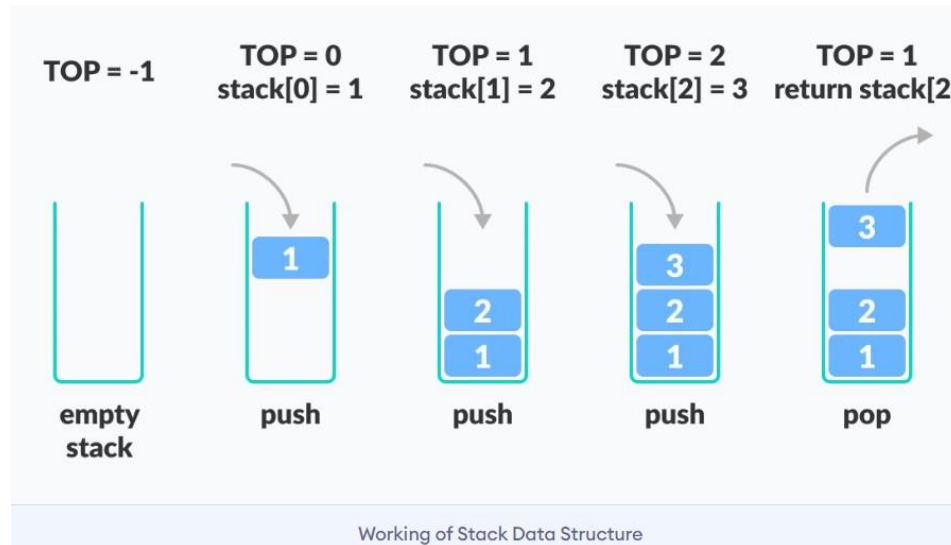
A. TUJUAN PRAKTIKUM

- 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 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 dimasuki



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.

C. GUIDED

Sourcecode

```
#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";
}

```

```

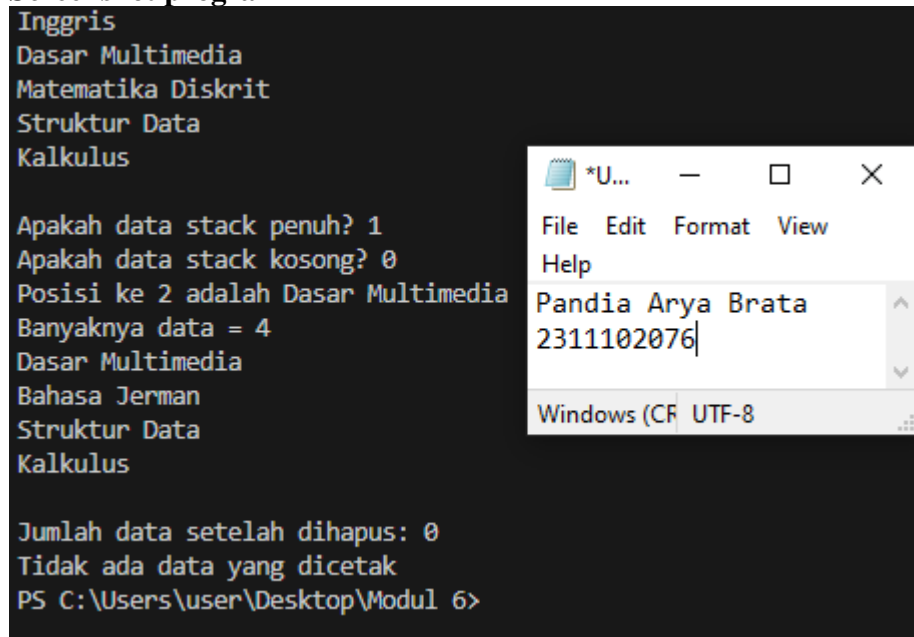
destroyArraybuku();
cout << "Jumlah data setelah dihapus: " << top << endl;

cetakArrayBuku();

return 0;
}

```

Screenshot program



Deskripsi program

Gambar diatas merupakan implementasi dari stack menggunakan array dalam bahasa C++. Stack merupakan struktur data yang menggunakan prinsip LIFO (Last In, First Out), artinya elemen terakhir yang dimasukkan ke dalam stack akan menjadi elemen pertama yang keluar. Kode tersebut mengimplementasikan struktur data stack menggunakan array. Data buku disimpan dalam array arrayBuku, dan variabel top digunakan untuk melacak posisi elemen teratas dalam stack. Kode tersebut menyediakan fungsi-fungsi dasar yang umum digunakan dalam operasi stack, seperti push (memasukkan elemen), pop (menghapus elemen), isEmpty (memeriksa apakah stack kosong), isFull (memeriksa apakah stack penuh), dan peek (melihat elemen pada posisi tertentu).

UNGUIDE

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.

Contoh:

Kalimat : ini

Kalimat tersebut adalah polindrom

Kalimat : telkom

Kalimat tersebut adalah bukan polindrom

```
Masukan Kalimat : ini
Kalimat tersebut adalah : Palindrom
```

Source Code :

```
#include <iostream>
#include <stack>
#include <string>
using namespace std;

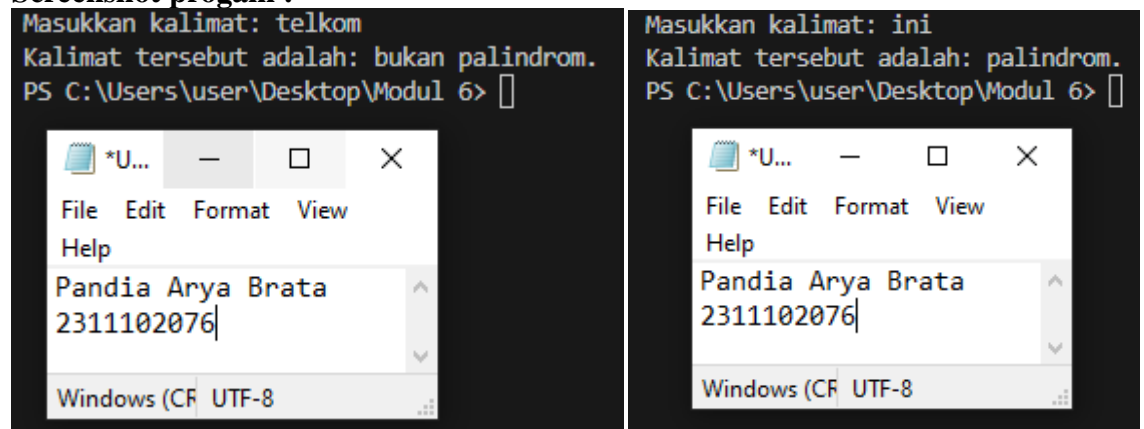
bool cekpalindrom(const string &kalimat)
{
    stack<char> STRUCTT;
    string kalimatkecil = "";
    // mengonversi semua huruf menjadi huruf kecil
    for (char abjad : kalimat)
    {
        kalimatkecil += tolower(abjad);
    }
    // Memasukkan semua karakter dari kalimat ke dalam stack
    for (char abjad : kalimatkecil)
    {
        STRUCTT.push(abjad);
    }
    // Memeriksa setiap karakter dari kalimat
    for (char abjad : kalimatkecil)
    {
        if (STRUCTT.top() != abjad)
        {
            return false;
        }
        STRUCTT.pop();
    }
    return true;
}
```

```

}
int main()
{
    string kalimat;
    cout << "Masukkan kalimat: ";
    getline(cin, kalimat);
    if (cekpalindrom(kalimat))
    {
        cout << "Kalimat tersebut adalah: palindrom." << endl;
    }
    else
    {
        cout << "Kalimat tersebut adalah: bukan palindrom." << endl;
    }
    return 0;
}

```

Screenshot program :



Deskripsi program :

Program ini menggunakan stack untuk mengecek apakah suatu kalimat merupakan palindrom dengan cara membandingkan karakter dari depan dan belakang kalimat. Stack membantu dalam menyimpan dan membandingkan karakter secara efisien.

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?

Contoh

Kalimat : Telkom Purwokerto

Hasil : otrekowruP mokleT

```
Masukkan Kata Telkom Purwokerto
Datastack Array :
Data : otrekowruP mokleT
```

Source Code :

```
#include <iostream>
#include <stack>
using namespace std;

string membalikKalimat(string kalimat)
{
    stack<char> charStack;
    string Hasil;

    for (int i = 0; i < kalimat.length(); i++)
    {
        charStack.push(kalimat[i]);
    }

    while (!charStack.empty())
    {
        Hasil += charStack.top();
        charStack.pop();
    }

    return Hasil;
}

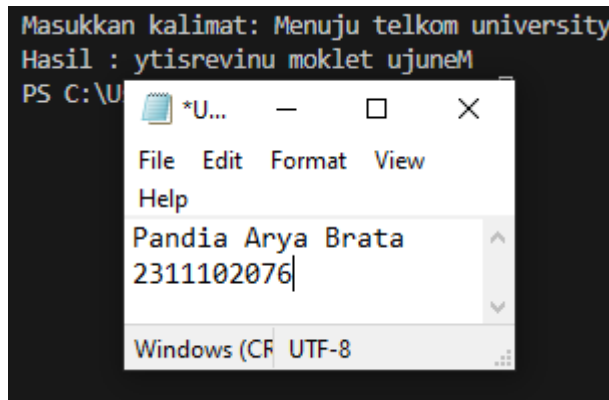
int main()
{
    string kalimat;

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

    string Hasil = membalikKalimat(kalimat);
    cout << "Hasil : " << Hasil << endl;

    return 0;
}
```

Screenshot program :



Deskripsi program :

Gambar diatas adalah program yang menggunakan stack untuk membalikkan sebuah kalimat Kode di atas menggunakan stack untuk membalikkan sebuah kalimat. Fungsi (***membalikKalimat***) menerima sebuah string sebagai argumen dan mengembalikan string hasil pembalikan kalimat. String yang diterima sebagai argumen diiterasi, dan setiap karakternya dimasukkan ke dalam stack. Setelah semua karakter dimasukkan ke dalam stack, karakter teratas stack diambil, ditambahkan ke string hasil, dan dihapus dari stack. Proses pengambilan karakter teratas dan penghapusan dilakukan hingga stack kosong. Di dalam 'main()', input kalimat dimasukkan melalui (***getline***) dan hasil pembalikan kalimat ditampilkan. Program berakhir dengan mengembalikan nilai 0 sebagai indikasi keberhasilan program.

D. DAFTAR PUSTAKA

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