LAPORAN PRAKTIKUM STRUKTUR DATA DAN ALGORITMA

MODUL 6

“STACK”

DISUSUN OLEH :

M.Fathiakmal.L.A

18102096

DOSEN

WAHYU ANDI SAPUTRA, S.PD., M.PD.

PROGRAM STUDI S1 TEKNIK INFORMATIKA

FAKULTAS INFORMATIKA

INSTITUT TEKNOLOGI TELKOM PURWOKERTO

2024

A. Dasar Teori

1. Stack

Stack adalah struktur data linier yang mengikuti aturan tertentu untuk melakukan operasi. Data yang memiliki struktur stack, tersusun seperti tumpukan, sehingga hanya elemen yang baru dimasukkan yang dapat diakses atau dilihat. Ujung tumpukan yang digunakan untuk melakukan semua operasi disebut bagian atas tumpukan. Stack mengikuti prinsip LIFO (Last In First Out), yang berarti elemen yang dimasukkan terakhir akan menjadi elemen pertama yang keluar dari urutan data.

Ujung tumpukan yang digunakan untuk melakukan semua operasi disebut bagian atas tumpukan. Stack mengikuti prinsip LIFO (Last In First Out), yang berarti elemen yang dimasukkan terakhir akan menjadi elemen pertama yang keluar dari urutan data.

2. Queue

Queue adalah struktur data linier di mana kita dapat menyisipkan dan menghapus elemen dari daftar data. Akhir daftar dari mana elemen disisipkan disebut ujung belakang dan ujung di mana elemen dihapus adalah ujung depan.

Struktur data yang menggunakan queue mengikuti prinsip FIFO (First In First Out), yang berarti elemen yang dimasukkan pertama kali dari ujung belakang akan menjadi elemen pertama yang dihapus dari ujung depan. Selain itu, terdapat dua istilah lain dalam queue, yakni operasi enqueue dan operasi dequeue. Operasi enqueue adalah teknik penyisipan pada struktur data queue, sedangkan operasi dequeue adalah teknik penghapusan pada struktur data queue.

Perbedaan Stack dan Queue

Terdapat beberapa perbedaan antara stack dan queue. Berikut beberapa perbedaannya.

- **Struktur Data**

  Queue adalah struktur data sederhana yang berfungsi sebagai antrean. Elemen yang ditempatkan pertama kali akan menjadi yang pertama kali diambil, mirip dengan antrean di toko atau bank. Sedangkan Stack adalah struktur data yang memungkinkan penambahan dan penghapusan elemen selalu di akhir atau ujung bawah dan atas, mirip dengan tumpukan buku yang diletakkan satu per satu.

- **Urutan**

  Pada Queue, urutan elemen harus dijaga agar elemen yang ditambahkan selalu ditempatkan di ujung belakang, sementara urutan elemen yang diambil dari depan. Namun pada Stack, urutan elemen kurang begitu penting, karena elemen yang ditambahkan terakhir akan selalu diambil terlebih dahulu.

- **Penggunaan**

  Queue digunakan pada aplikasi-antrian seperti algoritma BFS (Breadth First Search), dan job queue untuk mengatur antrian pekerjaan. Sedangkan penggunaan Stack lebih memfokuskan kepada pemanggilan fungsi atau prosedur dalam aplikasi, seperti Undo dan Redo di aplikasi pengolah teks.

B. Guided

Guided 1

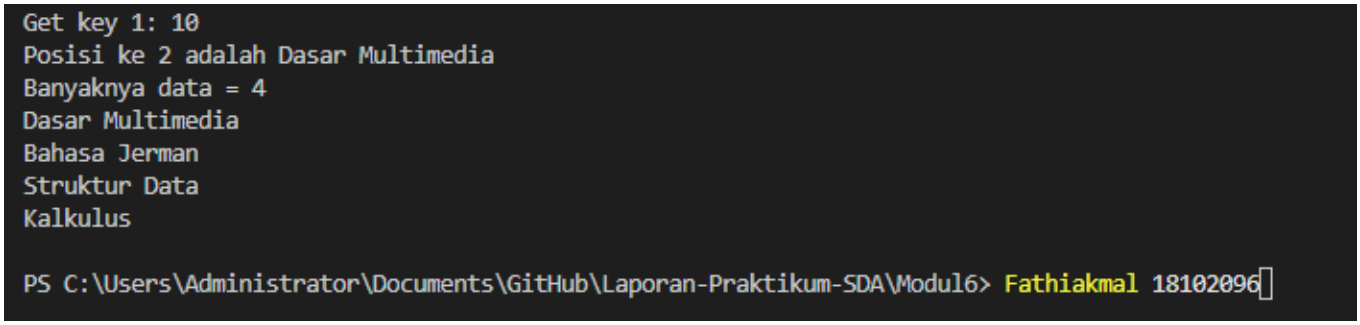
Source Code :

```
#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";
}
```

Screenshots Output:



Deskripsi Program :

Program di atas mengimplementasikan stack menggunakan array untuk menyimpan judul buku dalam bahasa C++. Stack ini memiliki kapasitas maksimal 5 buku yang diimplementasikan dengan array arrayBuku. Program ini mencakup beberapa fungsi seperti isFull untuk memeriksa apakah stack penuh, isEmpty untuk memeriksa apakah stack kosong, pushArrayBuku untuk menambahkan buku ke stack, popArrayBuku untuk menghapus buku dari stack, peekArrayBuku untuk melihat buku pada posisi tertentu, countStack untuk menghitung jumlah buku dalam stack, changeArrayBuku untuk mengganti buku pada posisi tertentu, destroyArraybuku untuk menghapus semua buku dalam stack, dan cetakArrayBuku untuk mencetak semua buku dalam stack. Fungsi main mendemonstrasikan penggunaan fungsi-fungsi ini dengan menambahkan, menghapus, melihat, mengganti, menghitung, dan mencetak buku dalam stack.

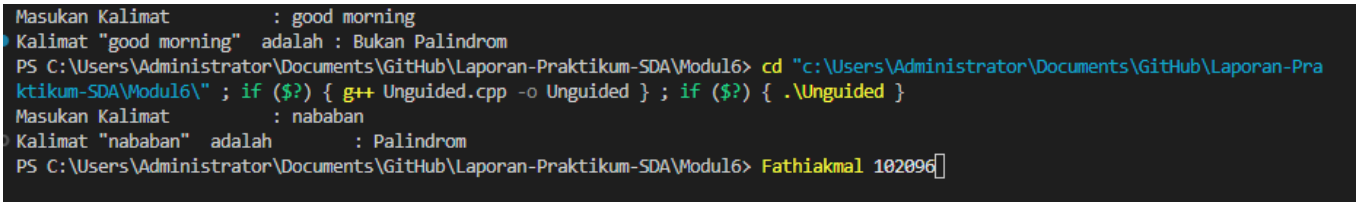
C. Unguided

1. 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 << "Masukan Kalimat\t\t : ";
    getline(cin, input);

    if (isPalindrome(input)) {
```

```
        cout << "Kalimat \"" << input << "\"  adalah\t : Palindrom" << endl;
    } else {
        cout << "Kalimat \"" << input << "\"  adalah : Bukan Palindrom" << endl;
    }

    return 0;
}
```

Screenshot output



Deskripsi Program :

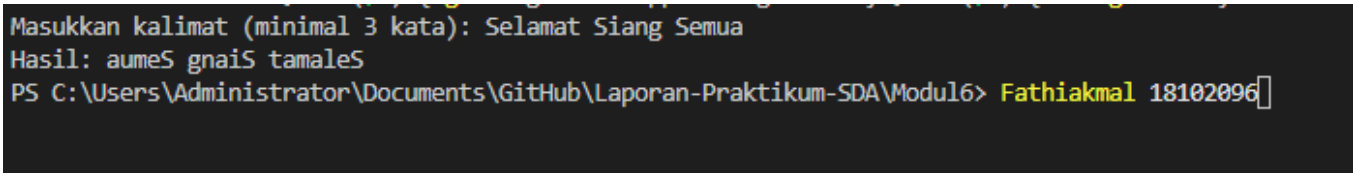
Program di atas memeriksa apakah sebuah string merupakan palindrom atau bukan, mengabaikan karakter non-huruf dan tidak peka huruf besar/kecil. Pertama, program mengimpor pustaka yang diperlukan, kemudian mendefinisikan fungsi isPalindrome yang menerima string sebagai argumen. Fungsi ini menggunakan stack (std::stack) untuk memeriksa palindrom. Pertama, string dibersihkan dari karakter non-huruf dan diubah menjadi huruf kecil. Kemudian, karakter-karakter dari string yang telah dibersihkan dimasukkan ke dalam stack. Selanjutnya, karakter dari string yang telah dibersihkan dibandingkan dengan karakter di puncak stack satu per satu. Jika ada karakter yang tidak cocok, fungsi mengembalikan false, menunjukkan bahwa string tersebut bukan palindrom. Jika semua karakter cocok, fungsi mengembalikan true, menunjukkan bahwa string tersebut adalah palindrom. Dalam fungsi main, program meminta pengguna untuk memasukkan sebuah kalimat, kemudian memanggil isPalindrome untuk memeriksa apakah kalimat tersebut palindrom dan menampilkan hasilnya.

2. Source Kode

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

```cpp
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;
}
```

Screnshoot Output



Deskripsi Program

Program ini membalikkan urutan kata dalam sebuah kalimat yang diberikan oleh pengguna, asalkan kalimat tersebut terdiri dari minimal tiga kata. Pada awalnya, pustaka yang diperlukan diimpor dan fungsi reverseWords didefinisikan untuk membalikkan urutan karakter dalam string yang diterima sebagai argumen. Fungsi ini menggunakan stack (std::stack) untuk menyimpan setiap karakter dalam string, kemudian mengosongkan stack tersebut untuk membentuk string yang dibalik. Dalam fungsi main, program meminta pengguna untuk memasukkan sebuah kalimat dan memeriksa jumlah kata dalam kalimat tersebut. Jika kalimat memiliki kurang dari tiga kata, program menampilkan pesan kesalahan. Jika tidak, fungsi reverseWords dipanggil untuk membalikkan urutan karakter dalam kalimat, dan hasilnya ditampilkan kepada pengguna.

D. Kesimpulan

Kesimpulan dari materi tentang stack adalah bahwa stack adalah struktur data yang mengikuti prinsip LIFO (Last In, First Out), di mana elemen terakhir yang dimasukkan adalah elemen pertama yang akan diambil. Stack menyediakan operasi dasar seperti push (menambahkan elemen ke atas stack), pop (menghapus elemen dari atas stack), dan top (mengakses elemen teratas tanpa menghapusnya).

Penggunaan stack sangat berguna dalam berbagai aplikasi seperti pemrosesan ekspresi aritmatika, penjagaan status eksekusi dalam pemrograman rekursif, dan algoritma penelusuran graf seperti DFS (Depth First Search). Stack juga sering digunakan untuk membalikkan data, seperti dalam contoh program di atas, di mana stack digunakan untuk membalikkan karakter dalam string.

Secara keseluruhan, stack adalah alat yang efektif untuk manajemen data dengan urutan tertentu, terutama ketika dibutuhkan akses hanya pada elemen terbaru.

E. Referensi

[1] Asisten Pratikum “Modul 6 Stack”, Learning Management System, 2024.

[2] Kompas. Pengertian Stack dan Queue. 1 Desember 2022.

Diakses Pada 19 Mei 2024, dari

Pengertian Stack dan Queue serta Contoh Penerapannya (kompas.com)

[3] Localstartupfest (2023, juli). “Perbedaan Queue dan Stack”.

Diakses pada 19 Mei 2024, dari

Perbedaan Queue dan Stack: Penjelasan Lengkap Dalam Ilmu Komputer - Localstartupfest.id