

LAPORAN PRAKTIKUM

MODUL VI STACK



Disusun oleh:
FATTAH RIZQY ADHIPRATAMA
NIM: 2311102019

Dosen Pengampu:
Wahyu Andi Saputra, S.Pd., M.Eng.

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

BAB I

TUJUAN PRAKTIKUM

1. Mampu memahami konsep stack pada struktur data dan algoritma
2. Mampu mengimplementasikan operasi-operasi pada stack
3. Mampu memecahkan permasalahan dengan solusi stack

BAB II

DASAR TEORI

Stack atau tumpukan adalah salah satu jenis struktur data pada komputer yang berfungsi untuk menyimpan dan mengorganisir data dalam urutan tertentu. Konsep utama dari stack adalah prinsip Last In First Out (LIFO), yang artinya data yang terakhir dimasukkan ke dalam stack akan menjadi data yang pertama kali diambil atau dikeluarkan dari stack.

Secara visual, stack dapat dibayangkan seperti tumpukan buku yang diletakkan satu per satu, di mana buku terakhir yang diletakkan akan menjadi buku yang paling mudah diambil atau diambil terlebih dahulu. Dalam implementasi pada program komputer, data pada stack disimpan pada satu tempat tertentu dalam memori komputer dan dapat diakses menggunakan operasi push (menambahkan data ke dalam stack) dan pop (mengeluarkan data dari stack).

Stack sering digunakan dalam implementasi algoritma dan fungsi matematis seperti evaluasi ekspresi matematis dan konversi infix ke postfix. Selain itu, stack juga digunakan dalam aplikasi yang membutuhkan manajemen tumpukan data seperti pengembangan sistem operasi, kompilator, dan perangkat lunak grafis.

Ada 2 jenis stack :

1. Stack Statis

Stack ini didefinisikan dengan ukuran yang tetap dan tidak dapat diubah selama proses eksekusi. Elemen dapat ditambahkan atau dihapus dari stack, tetapi ukurannya tetap sama. Stack statis dapat diimplementasikan dengan menggunakan array atau pointer.

2. Stack Dinamis

Stack ini didefinisikan dengan ukuran yang dapat diubah sesuai dengan jumlah elemen yang ada di dalamnya. Saat jumlah elemen di dalam stack melebihi kapasitas yang telah ditentukan, kapasitas stack akan ditambahkan secara

otomatis. Stack dinamis dapat diimplementasikan dengan menggunakan linked list atau array dinamis.

Fungsi-fungsi stack :

1. Pengelolaan memori: Stack digunakan untuk menyimpan alamat memori dari suatu fungsi saat program dijalankan.
2. Pengecekan tata bahasa: Stack digunakan untuk menyimpan konteks dari suatu ekspresi matematika atau kondisi logika.
3. Algoritma backtracking: Stack digunakan untuk menyimpan state dari suatu proses yang dapat dikembalikan ke state sebelumnya jika proses tersebut gagal.
4. Algoritma pengurutan: Stack digunakan dalam beberapa algoritma sorting seperti quicksort dan mergesort.
5. Pemecahan masalah rekursif: Stack dapat digunakan dalam pemecahan masalah rekursif karena setiap panggilan rekursif akan ditambahkan ke stack dan setiap pengembalian dari rekursif akan dihapus dari stack.

BAB III

GUIDED

1. 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";

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

    cetakArrayBuku();

    return 0;
}

```

Screenshoot program

```
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 6>
code.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher
1fk.gkw' '--stdout=Microsoft-MIEngine-Out-h5jwr1r2.ubx' '--stderr=Micro
=Microsoft-MIEngine-Pid-wqcq4lxb.2iq' '--dbgExe=C:\msys64\ucrt64\bin\
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 D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 6>
```

Deskripsi program

Program ini merupakan implementasi dari sebuah stack. Stack adalah struktur data yang bekerja dengan prinsip LIFO (Last In, First Out). Program ini memiliki fungsi-fungsi dasar stack seperti push, pop, peek, isEmpty, isFull, countStack, changeArrayBuku, dan destroyArrayBuku. Program ini juga dapat mencetak isi dari stack, memeriksa apakah stack penuh atau kosong, melihat elemen pada posisi tertentu, menghitung jumlah elemen dalam stack, mengubah data pada posisi tertentu, serta menghapus seluruh data dalam stack.

LATIHAN KELAS - UNGUIDED

1. Unguided 1

Source code

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

using namespace std;

bool isPalindrome(string kalimat) {
    //Konversi kalimat menjadi huruf kecil
    for (int i = 0; i < kalimat.length(); i++) {
        kalimat[i] = tolower(kalimat[i]);
    }

    //Hapus spasi dan karakter non-huruf
    string temp = "";
    for (int i = 0; i < kalimat.length(); i++) {
        if (isalpha(kalimat[i])) {
            temp += kalimat[i];
        }
    }

    //Buat stack untuk menyimpan karakter kalimat
    stack<char> s;
    for (int i = 0; i < temp.length(); i++) {
        s.push(temp[i]);
    }

    //Bandingkan karakter dari depan dan belakang
    for (int i = 0; i < temp.length() / 2; i++) {
        if (s.top() != temp[i]) {
            return false;
        }
        s.pop();
    }

    return true;
}
```

```

int main() {
    string kalimat;

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

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

    return 0;
}

```

Screenshoot program

```

PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 6>
code.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher
05m.bji' '--stdout=Microsoft-MIEngine-Out-xw0zg2qe.1aj' '--stderr=Mi
=Microsoft-MIEngine-Pid-0ncqqdyq.vyf' '--dbgExe=C:\msys64\ucrt64\bin
Masukkan kalimat: ini
Kalimat ini adalah palindrom
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 6>

```

```

PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 6>
Data dan Algoritme\Modul 6\" ; if ($?) { g++ tempCodeRunnerFile.cpp
odeRunnerFile }
Masukkan kalimat: telkom
Kalimat telkom bukan palindrom
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 6>

```

Deskripsi Program

Program ini adalah sebuah program sederhana untuk mengecek apakah sebuah kalimat merupakan palindrom atau tidak. Sebuah palindrom adalah sebuah kata, frasa, angka, atau susunan lainnya yang dapat dibaca dengan sama baik dari depan maupun dari belakang. Program ini salah satunya menggunakan fungsi "isPalindrome" yaitu fungsi yang digunakan untuk menerima sebuah string (kalimat) sebagai input dan mengembalikan nilai boolean true jika kalimat tersebut merupakan palindrom, dan false jika tidak.

2. Unguided 2

Source Code

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

using namespace std;

//Fungsi untuk membalikkan huruf-huruf dalam sebuah kalimat
menggunakan stack
string reverseLetters(const string& sentence) {
    stack<char> charStack;

    //Memasukkan setiap huruf ke dalam stack
    for (char c : sentence) {
        charStack.push(c);
    }

    //Membangun kalimat terbalik dari stack
    string reversedSentence;
    while (!charStack.empty()) {
        reversedSentence += charStack.top();
        charStack.pop();
    }

    return reversedSentence;
}

int main() {
    string sentence;
    cout << "Masukkan kalimat (minimal 3 kata) : ";
    getline(cin, sentence);

    string reversed = reverseLetters(sentence); // Balikkan
    huruf-huruf dalam kalimat dari paling akhir ke awal menggunakan
    stack

    cout << "Hasil pembalikan kalimat : " << reversed << endl;

    return 0;
}
```

```
}
```

Screenshoot Program

```
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 6>  
code.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher  
nwv.lji' '--stdout=Microsoft-MIEngine-Out-gtlzleey.nad' '--stderr=Mic  
=Microsoft-MIEngine-Pid-ggunstrt.y0n' '--dbgExe=C:\msys64\ucrt64\bin\  
Masukkan kalimat (minimal 3 kata) : nama saya Fattah Rizqy  
Hasil pembalikan kalimat : yqziR hattaF ayas aman  
PS D:\Data Semester 2\Praktikum Struktur Data dan Algoritme\Modul 6>
```

Deskripsi Program

Program ini adalah sebuah program sederhana yang membalikkan huruf-huruf dalam sebuah kalimat menggunakan stack. Program ini menggunakan struktur data stack untuk membalikkan urutan huruf-huruf dalam kalimat. Dengan memasukkan setiap huruf ke dalam stack dan kemudian mengambil huruf-huruf tersebut secara berurutan dari stack, program dapat menghasilkan kalimat yang huruf-hurufnya telah dibalik.

BAB IV

KESIMPULAN

Stack atau tumpukan adalah struktur data yang bekerja dengan prinsip LIFO (Last In, First Out), di mana elemen yang terakhir dimasukkan adalah yang pertama kali dikeluarkan. Stack sering dianalogikan dengan tumpukan piring, di mana piring yang terakhir ditaruh di atas adalah yang pertama kali diambil.

Terdapat beberapa operasi dasar yang dapat dilakukan pada stack, seperti push (untuk menambahkan elemen ke atas stack), pop (untuk menghapus elemen teratas dari stack), dan peek (untuk melihat elemen teratas tanpa menghapusnya). Operasi lainnya meliputi isEmpty (untuk memeriksa apakah stack kosong) dan isFull (untuk memeriksa apakah stack penuh pada implementasi menggunakan array).

Stack merupakan struktur data yang penting dan memiliki banyak aplikasi dalam pemrograman. Praktikum struktur data stack membantu mahasiswa untuk memahami konsep stack, operasi dasar stack, dan aplikasinya dalam berbagai bidang.

DAFTAR PUSTAKA

- [1] Sutiono, 2023. Struktur Data Stack: Pengertian, Jenis dan Contoh. Diakses pada tanggal 15 Mei 2024, dari <https://dosenit.com/ilmu-komputer/struktur-data-stack>
- [2] LamanIT, 2023. Struktur Data Stack: Pengertian, Fungsi, Contoh dan Kelebihan. Diakses pada tanggal 15 Mei 2024, dari <https://lamanit.com/struktur-data-stack/>