

**LAPORAN PRAKTIKUM
STRUKTUR DATA DAN ALGORITMA**

**MODUL VI
STACK**



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

Disusun oleh:

HAIKAL SATRIATAMA

2311102066

IF-11-B

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

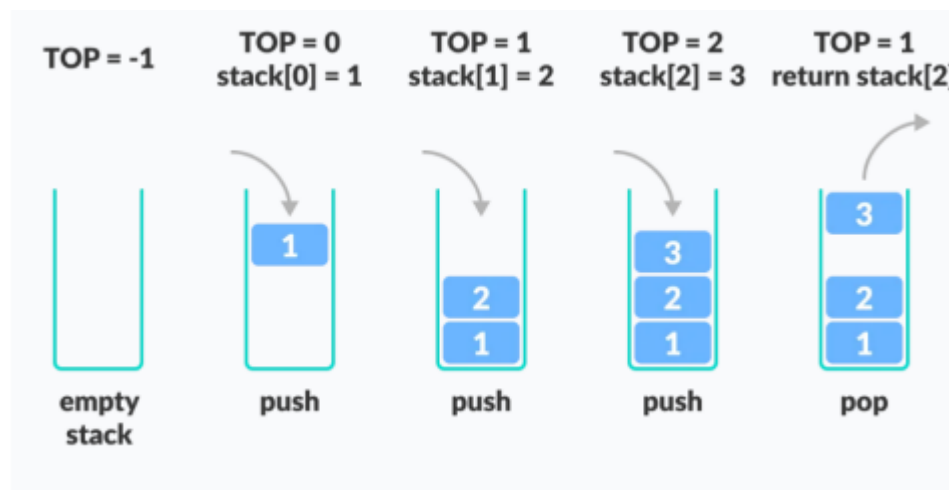
2024

BAB I

DASAR TEORI

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 terurut 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.

g. Peek (Lihat): Melihat nilai atau elemen pada posisi tertentu dalam tumpukan tanpa menghapusnya.

h. Clear (Hapus Semua): Mengosongkan atau menghapus semua elemen dari tumpukan.

i. Search (Cari): Mencari keberadaan elemen tertentu dalam tumpukan.

BAB II

GUIDED

LATIHAN – 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

The screenshot shows a C++ program running in a terminal window. The program manages a stack of books. The output is as follows:

```

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

```

The terminal window also shows a status bar at the bottom: "Ln 2, Col 17 | 43 karakter | 100% | Window UTF-8".

Deskripsi program

- **isFull():** Memeriksa apakah tumpukan sudah penuh.
- **isEmpty():** Memeriksa apakah tumpukan kosong.
- **pushArrayBuku(string data):** Menambahkan buku baru ke tumpukan.
- **popArrayBuku():** Menghapus buku terakhir dari tumpukan.
- **peekArrayBuku(int posisi):** Melihat buku pada posisi tertentu dalam tumpukan.
- **countStack():** Menghitung jumlah buku dalam tumpukan.
- **changeArrayBuku(int posisi, string data):** Mengubah buku pada posisi tertentu dalam tumpukan.
- **destroyArraybuku():** Menghapus semua buku dari tumpukan.

- **cetakArrayBuku():** Mencetak semua buku dalam tumpukan.

BAB III

UNGUIDED

TUGAS – UNGUIDED

1. 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 <string>

using namespace std;

// Fungsi untuk menghapus karakter non-alfabet dari string
string removeNonAlphanumeric(string str) {
    string result = "";
    for (char c : str) {
        if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z')) {
            result += tolower(c); // Ubah huruf menjadi lowercase
        }
    }
    return result;
}

// Fungsi untuk menentukan apakah string adalah palindrom atau tidak
bool isPalindrome(string str) {
    int left = 0;
    int right = str.length() - 1;

    while (left < right) {
        // Lewati karakter non-alfabet di kiri
        while (left < right && !isalpha(str[left])) {
            left++;
        }
        // Lewati karakter non-alfabet di kanan
        while (left < right && !isalpha(str[right])) {
            right--;
        }
        // Periksa apakah karakter di kiri sama dengan karakter di
        kanan
        if (tolower(str[left]) != tolower(str[right])) {
            return false;
        }
        left++;
        right--;
    }
}
```



```

    }

    return true;
}

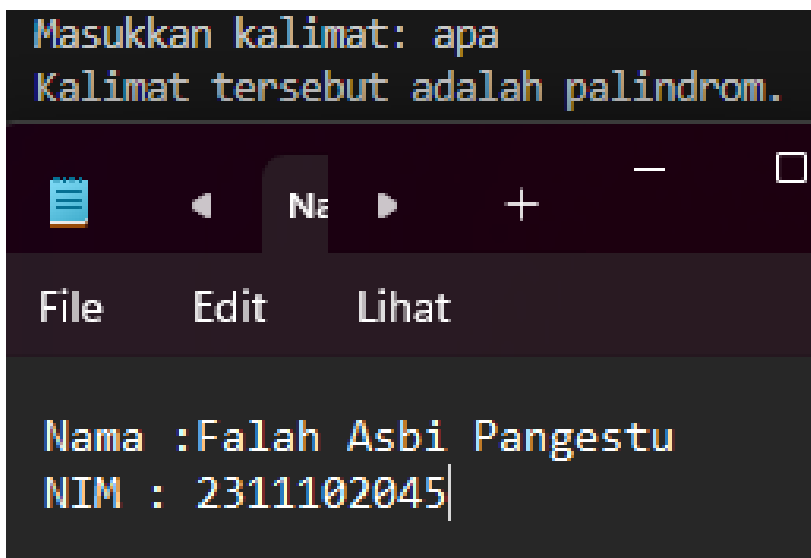
int main() {
    string kalimat;
    cout << "Masukkan kalimat: ";
    getline(cin, kalimat);

    string cleanedString = removeNonAlphanumeric(kalimat);
    if (isPalindrome(cleanedString)) {
        cout << "Kalimat tersebut adalah palindrom." << endl;
    } else {
        cout << "Kalimat tersebut bukan palindrom." << endl;
    }

    return 0;
}

```

Screenshot Program



Deskripsi program

- `removeNonAlphanumeric(string str)`: Fungsi ini menerima string sebagai input dan mengembalikan string baru yang hanya berisi karakter alfanumerik (huruf dan angka).
- `isPalindrome(string str)`: Fungsi ini menerima string sebagai input dan mengembalikan `true` jika string tersebut palindrom, dan `false` jika bukan.

Cara kerja :

- Pengguna memasukkan kalimat, misalnya “apa”

- Program akan menghapus karakter non-alfabet dan mengubah huruf menjadi lowercase.
- Setelah membersihkan kalimat, program akan memeriksa apakah “apa” adalah palindrom. Jika iya, program akan mencetak “Kalimat tersebut adalah palindrom.”

2. Guided 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;

// Fungsi untuk membalikkan kalimat menggunakan stack
string reverseSentence(string sentence) {
    stack<char> charStack;
    string reversedSentence = "";

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

    // Pop setiap karakter dari stack untuk mendapatkan kalimat
    terbalik
    while (!charStack.empty()) {
        reversedSentence += charStack.top();
        charStack.pop();
    }

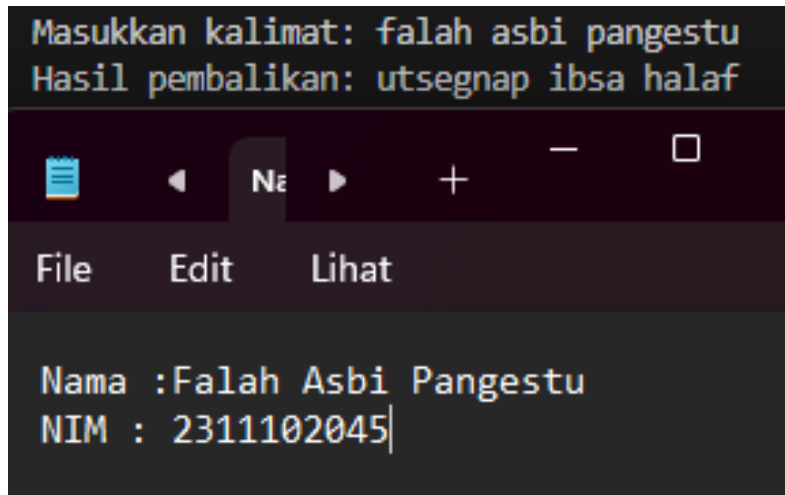
    return reversedSentence;
}

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

    string hasilPembalikan = reverseSentence(kalimat);
    cout << "Hasil pembalikan: " << hasilPembalikan << endl;
```

```
    return 0;  
}
```

Screenshoot program



Deskripsi program

- Meminta input: Mencetak prompt "Masukkan kalimat: " dan menggunakan `getline(cin, kalimat)` untuk mengambil kalimat dari pengguna dan menyimpannya dalam variabel `kalimat`.
- Membalikkan kalimat: Memanggil fungsi `reverseSentence(kalimat)` dan menyimpan hasilnya dalam variabel `hasilPembalikan`.
- Menampilkan hasil: Mencetak "Hasil pembalikan: " dan kemudian menampilkan nilai `hasilPembalikan` (kalimat yang dibalik).
- Mengembalikan nilai: Mengembalikan nilai 0 untuk menunjukkan bahwa program telah selesai dengan sukses.

BAB IV

KESIMPULAN

Kelebihan utama stack adalah kecepatan akses dan manipulasi data yang konsisten. Operasi-operasi pada stack memiliki kompleksitas waktu $O(1)$, artinya operasi push, pop, dan top dilakukan dalam waktu konstan, terlepas dari ukuran stack. Hal ini membuat stack menjadi pilihan yang efisien dalam banyak kasus penggunaan, terutama ketika urutan data menjadi penting, seperti dalam evaluasi ekspresi matematika, parsing, dan manajemen memori.

Namun, stack juga memiliki kelemahan. Kapasitas stack terbatas oleh ukuran memori yang tersedia. Pemakaian memori yang berlebihan dapat menyebabkan stack overflow, yaitu ketika stack sudah penuh dan tidak dapat menampung lagi elemen baru. Jadi, perlu berhati-hati dalam penggunaan stack agar tidak mengalami masalah ini

DAFTAR PUSTAKA

- [1] Trivusi web, Struktur data Stack : Pengertian, karakteristik, dan jenis-jenisnya (2022) diakses pada 19 Mei 2024 <https://www.trivusi.web.id/2022/07/struktur-data-stack.html>
- [2] Learning Management System. Asisten Praktikum. Modul 6 Stack. 2024