

LAPORAN PRAKTIKUM

MODUL VI STACK



Disusun oleh:
Imelda Fajar Awalina Crisyanti
NIM: 2311102004

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

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
PURWOKERTO
2023**

BAB I

TUJUAN PRAKTIKUM

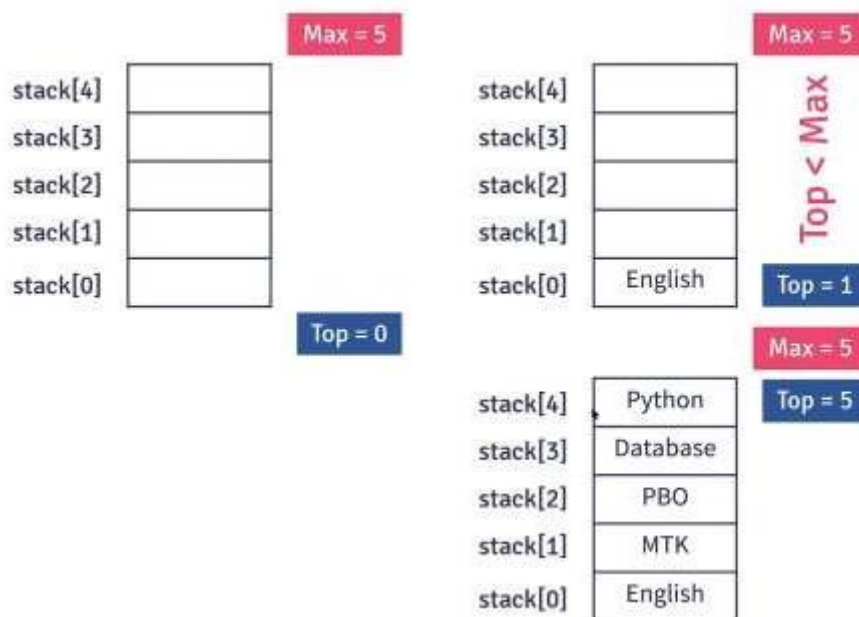
- a. Mampu memahami konsep stack pada struktur data dan algoritma
- b. Mampu mengimplementasikan operasi-operasi pada stack
- c. Mampu memecahkan permasalahan dengan solusi stack

BAB II

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.

- c. Top (Atas): Mendapatkan nilai atau melihat elemen teratas pada tumpukan tanpa menghapusnya.
- d. IsEmpty (Kosong): Memeriksa apakah tumpukan kosong atau tidak.
- e. IsFull (Penuh): Memeriksa apakah tumpukan penuh atau tidak (terutama pada implementasi tumpukan dengan kapasitas terbatas).
- f. 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 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

```
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
```

Deskripsi program

- **isFull()**: Mengembalikan nilai true jika tumpukan sudah penuh (terisi 5 buku) dan false jika masih ada ruang.
- **isEmpty()**: Mengembalikan nilai true jika tumpukan kosong (tidak ada buku) dan false jika ada buku di tumpukan.
- **pushArrayBuku(string data)**: Menambahkan buku baru (data) ke tumpukan.
- **popArrayBuku()**: Menghapus buku teratas dari tumpukan.
- **peekArrayBuku(int posisi)**: Menampilkan judul buku pada posisi tertentu (dimulai dari 1) di tumpukan.

- **countStack():** Menghitung jumlah buku yang ada di tumpukan.
- **changeArrayBuku(int posisi, string data):** Mengubah judul buku pada posisi tertentu (dimulai dari 1) di tumpukan dengan data baru.
- **destroyArrayBuku():** Menghapus semua buku dari tumpukan.
- **cetakArrayBuku():** Menampilkan daftar judul buku di tumpukan dari atas ke bawah.

Program mendeklarasikan array untuk menyimpan judul buku (maksimal 5 buku) dan variabel untuk melacak posisi teratas di tumpukan. Lima buku ditambahkan ke tumpukan: "Kalkulus", "Struktur Data", "Matematika Diskrit", "Dasar Multimedia", dan "Inggris". Daftar buku di tumpukan dicetak. Program memeriksa apakah tumpukan penuh dan kosong. Judul buku pada posisi 2 ditampilkan. Buku teratas dihapus dari tumpukan. Jumlah buku di tumpukan dicetak. Judul buku pada posisi 2 diubah menjadi "Bahasa Jerman". Daftar buku di tumpukan dicetak. Semua buku dihapus dari tumpukan. Jumlah buku di tumpukan setelah dihapus dicetak. Daftar buku di tumpukan dicetak (yang kosong karena semua buku telah dihapus).

LATIHAN KELAS - UNGUIDED

1. Unguided 1

Source code

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

bool isPalindrome(string kalimat) {
    string KalimatHurufKecil = "";
    for (char c : kalimat) {
        if (isalpha(c)) {
            KalimatHurufKecil += tolower(c);
        }
    }

    stack<char> s;

    for (char c : KalimatHurufKecil) {
        s.push(c);
    }

    int i = 0;
    while (!s.empty() && i < KalimatHurufKecil.length()) {
        if (s.top() != KalimatHurufKecil[i]) {
            return false;
        }
        s.pop();
        i++;
    }

    return true;
}

int main() {
    string kalimat;

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

```

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

    return 0;
}

```

Screenshoot program

```

Masukkan kalimat: kasur ini rusak
kasur ini rusak adalah kalimat palindrom.

```

Deskripsi program

- **Membuat Fungsi isPalindrome:**
 - Mengubah kalimat menjadi huruf kecil.
 - Menyimpan karakter kalimat ke dalam tumpukan.
 - Memeriksa karakter dari awal dan akhir kalimat secara berurutan.
 - Jika ada ketidakcocokan, kalimat bukan palindrom.
 - Jika semua karakter cocok, kalimat merupakan palindrom.
- **Fungsi Utama (main):**
 - Meminta pengguna untuk memasukkan kalimat.
 - Memanggil fungsi isPalindrome untuk menentukan apakah kalimat palindrom.
 - Menampilkan pesan sesuai dengan hasil.

2. Unguided 2

Source code

```

#include <iostream>
#include <stack>

using namespace std;

void KalimatTerbalik (string kalimat) {

```

```

stack<char> s;

for (char c : kalimat ) {
    s.push(c);
}

string KalimatTerbalik = "";

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

cout << "Kalimat terbalik: " << KalimatTerbalik << endl;
}

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

    if (kalimat.size() < 3) {
        cout << "Kalimat harus minimal 3 kata!" << endl;
        return 1;
    }

    KalimatTerbalik (kalimat);

    return 0;
}

```

- Program ini menggunakan stack untuk menyimpan karakter kalimat secara terbalik.
- Karakter-karakter tersebut kemudian dikeluarkan dari stack secara berurutan untuk membentuk kalimat terbalik.
- Pastikan kalimat yang dimasukkan minimal 3 kata.

BAB IV

KESIMPULAN

Inti dari stack adalah prinsip LIFO (Last In, First Out) [konsep LIFO stack], dimana data yang terakhir masuk menjadi yang pertama keluar.

Operasi dasar pada stack adalah push dan pop. Push berfungsi untuk memasukkan data ke bagian atas stack, sedangkan pop untuk mengeluarkan data dari bagian atas stack [operasi stack].

Praktikum ini kemungkinan besar mengajarkan bagaimana membuat program untuk melakukan operasi push dan pop pada stack, mungkin menggunakan array atau linked list untuk menyimpan data stack. Selain itu, praktikum ini mungkin juga mencakup pengecekan kondisi stack kosong dan penuh.