

LAPORAN PRAKTIKUM

MODUL 6 STACK



Disusun oleh:
Nofita Fitriyani
NIM: 2311102001

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

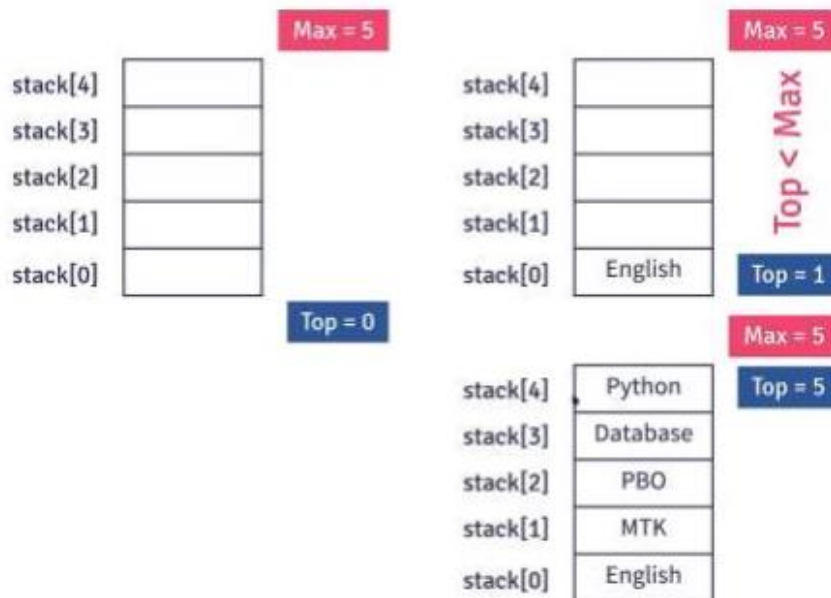
- a. Mampu memahami konsep stack pada struktur data dan algoritma
- b. Mampu mengimplementasikan operasi-operasi pada stack
- c. Mampu memecahkan permasalahan dengan solusi stack

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.
- 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).

- 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

```

e }; if ($?) { .\tempCodeRunnerFile }
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 C:\Praktikum Struktur Data\Modul 6>

```


Deskripsi program

Program ini merupakan implementasi stack menggunakan array di C++ untuk menyimpan judul buku dengan kapasitas maksimal lima buku. Fungsi utama meliputi `pushArrayBuku` untuk menambah buku, `popArrayBuku` untuk menghapus buku teratas, dan `peekArrayBuku` untuk melihat buku pada posisi tertentu dari atas stack. Fungsi `isFull` dan `isEmpty` digunakan untuk memeriksa apakah stack penuh atau kosong. Fungsi `countsStack` mengembalikan jumlah buku dalam stack, `changeArrayBuku` mengubah judul buku pada posisi tertentu, dan `destroyArrayBuku` menghapus semua buku dalam stack. Fungsi `cetakArrayBuku` mencetak seluruh buku dari yang teratas hingga terbawah. Pada fungsi `main`, program menambahkan lima buku, mencetaknya, memeriksa kondisi penuh atau kosong, melihat buku pada posisi kedua, menghapus buku teratas, mengubah judul buku pada posisi kedua, mencetak ulang, dan akhirnya menghapus semua buku serta mencetak stack yang sudah kosong.

LATIHAN KELAS - UNGUIDED

1. Unguided 1

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;

char stackData[77];
int top = -1;

bool isFull(int n) {
    return (top == n - 1);
}

bool isEmpty() {
    return (top == -1);
}

void push(char x) {
    if (!isFull(77)) {
        stackData[++top] = x;
    }
}

char pop() {
    if (isEmpty())
    {
        cout << "Tidak ada data yang dihapus" << endl;
    }
    else
    {
        return stackData[top--];
    }
}
```

```

    }
}

bool Palindrom(string input) {
    int n = input.length();
    for (int i = 0; i < n; i++) {
        push(input[i]);
    }
    string reversedStr = "";
    while (!isEmpty()) {
        reversedStr += pop();
    }
    return (input == reversedStr);
}

int main() {
    string input;
    cout << "Masukan Kalimat : ";
    getline(cin, input);
    cout << endl;
    cout << "Kalimat : " << input << endl;
    if (Palindrom(input)) {
        cout << "Kalimat tersebut adalah kalimat Palindrom" <<
endl;
    } else {
        cout << "Kalimat tersebut adalah bukan Palindrom"<< endl;
    }
    return 0;
}

```

Screenshoot program

```

Masukan Kalimat : kasur ini rusak

Kalimat : kasur ini rusak
Kalimat tersebut adalah kalimat Palindrom
PS C:\Praktikum Struktur Data\Modul 6>

```

Deskripsi program

Program ini memeriksa apakah sebuah string yang dimasukkan pengguna adalah palindrom menggunakan stack di C++. String diinput oleh pengguna dan disimpan dalam array stackData dengan top sebagai indeks teratas stack. Fungsi isFull dan isEmpty memeriksa apakah stack penuh atau kosong. Fungsi push menambahkan karakter ke stack jika tidak penuh, sedangkan pop menghapus dan mengembalikan palindrom mengisi stack dengan karakter string, membalik string dengan mengeluarkan karakter dari stack, lalu membandingkan string asli dengan string yang dibalik untuk menentukan apakah string tersebut palindrom. Hasil pemeriksaan ditampilkan ke pengguna.

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 <string>
using namespace std;

char stackData[77];
int top = -1;

bool isFull(int n) {
    return (top == n - 1);
}

bool isEmpty() {
    return (top == -1);
}

void push(char x) {
    if (!isFull(77)) {
        stackData[++top] = x;
    }
}
```

```

    }
}
char pop() {
    return stackData[top--];
}
bool reverse(string input) {
    int n = input.length();
    for (int i = 0; i < n; i++) {
        push(input[i]);
    }
    string reversedStr = "";
    while (!isEmpty()) {
        reversedStr += pop();
    }
    cout<< "Hasil          : "<< reversedStr;
}

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

    return 0;
}

```

Screenshoot program

```

Masukan kalimat : Nofita Bau Wangi
Hasil          : ignaW uaB atifoN
PS C:\Praktikum Struktur Data\Modul 6>

```

Deskripsi program

Program ini membalik string yang dimasukkan pengguna menggunakan stack. String disimpan dalam array `stackData` dengan `top` sebagai indeks teratas. Fungsi `isFull` memeriksa apakah stack penuh, dan `isEmpty` memeriksa apakah stack kosong. Fungsi `push` menambahkan karakter ke stack jika belum penuh, dan `pop` menghapus serta mengembalikan karakter teratas. Fungsi `reverse` membalik string dengan memasukkan setiap karakter ke dalam stack, lalu mengeluarkannya satu per satu untuk membentuk string yang dibalik. Program meminta pengguna memasukkan string, membaliknya menggunakan `reverse`, dan menampilkan hasilnya.

BAB IV

KESIMPULAN

Kesimpulannya adalah bahwa stack merupakan struktur data sederhana yang beroperasi berdasarkan prinsip Last In First Out (LIFO), di mana elemen terakhir yang dimasukkan akan menjadi elemen pertama yang dihapus. Stack mirip dengan tumpukan piring di mana piring terakhir yang ditambahkan berada di atas dan menjadi piring pertama yang diambil. Operasi dasar pada stack meliputi push (menambahkan elemen), pop (menghapus elemen teratas), top (melihat elemen teratas tanpa menghapusnya), isEmpty (memeriksa apakah stack kosong), isFull (memeriksa apakah stack penuh), size (menghitung jumlah elemen), peek (melihat elemen pada posisi tertentu), clear (menghapus semua elemen), dan search (mencari elemen tertentu dalam stack). Stack sangat berguna dalam berbagai aplikasi komputer, seperti manajemen memori, pengurutan, dan evaluasi ekspresi.