

LAPORAN PRAKTIKUM

MODUL 6 STACK



**Disusun oleh:
Rakha Yudhistira
NIM: 2311102010**

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 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:

- a. Push (Masukkan): Menambahkan elemen ke dalam tumpukan pada posisi paling atas atau ujung.
- b. 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

```

PS C:\IT\SEMESTER 1\Struktur Data\Tugas (code & dokumentasi)\code 1> g++ guided1.cpp -o guided1
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

Program ini adalah implementasi stack menggunakan array of string. Stack ini didefinisikan dalam variabel arrayBuku dengan kapasitas maksimum maksimal dan indeks atas top. Program ini memiliki beberapa fungsi untuk mengelola stack, yaitu:

1. `isFull()`: Fungsi ini digunakan untuk mengecek apakah stack penuh atau tidak.
2. `isEmpty()`: Fungsi ini digunakan untuk mengecek apakah stack kosong atau tidak.
3. `pushArrayBuku(string data)`: Fungsi ini digunakan untuk menambahkan data ke stack.
4. `popArrayBuku()`: Fungsi ini digunakan untuk mengambil data teratas dari stack.
4. `peekArrayBuku(int posisi)`: Fungsi ini digunakan untuk melihat data pada posisi tertentu dari stack tanpa mengambilnya.
5. `countStack()`: Fungsi ini digunakan untuk mengembalikan jumlah data yang ada di stack.
6. `changeArrayBuku(int posisi, string data)`: Fungsi ini digunakan untuk mengganti data pada posisi tertentu di stack.
7. `destroyArraybuku()`: Fungsi ini digunakan untuk menghapus semua data di stack.
8. `cetakArrayBuku()`: Fungsi ini digunakan untuk mencetak semua data yang ada di stack.

Di `main()`, program membuat stack dengan kapasitas maksimum 5 dan menambahkan beberapa data ke dalamnya. Program kemudian mencetak semua data yang ada di stack, mengecek apakah stack penuh atau kosong, melihat data pada posisi ke-2, mengambil data teratas dari stack, mengganti data pada posisi ke-2, dan menghapus semua data di stack. Program kemudian mencetak semua data yang ada di stack setelah dihapus.

LATIHAN KELAS - UNGUIDED

1. Unguided 1

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

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

Screenshoot program

```
PS D:\ITTP\Semester 2\Struktur dat
odul 6\" ; if ($?) { g++ unguided1
Kalimat : ini
Kalimat ini adalah Palindrom
PS D:\ITTP\Semester 2\Struktur dat
odul 6\" ; if ($?) { g++ unguided1
Kalimat : inu
Kalimat inu bukan Palindrom
PS D:\ITTP\Semester 2\Struktur dat
```

Deskripsi program

Program diatas adalah program yang digunakan untuk memeriksa apakah suatu kalimat atau teks merupakan palindrom atau bukan. Palindrom adalah teks yang sama jika dibaca dari kiri ke kanan maupun dari kanan ke kiri. Program ini menggunakan dua buah stack untuk menyimpan karakter-karakter dari teks yang dimasukkan. Pertama, semua karakter dari teks dimasukkan ke dalam stack. Kemudian, setiap karakter dari teks yang dimasukkan dibandingkan dengan karakter teratas dari stack. Jika karakter tersebut tidak sama, maka teks tersebut bukan palindrom. Jika semua karakter sama, maka teks tersebut adalah palindrom.

2. Unguided 2

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()
{

```

```

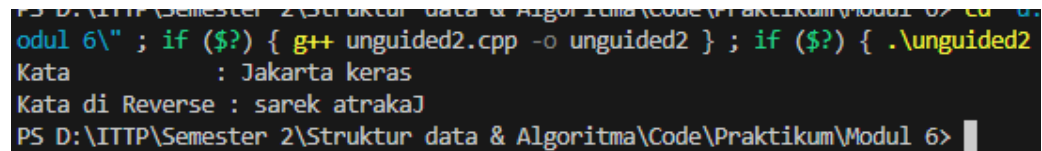
        return (top == -1);
    }
    void push(char x)
    {
        if (!isFull(77))
        {
            stackData[++top] = x;
        }
    }
    char pop()
    {
        return stackData[top--];
    }
    bool reverse(string str)
    {
        int n = str.length();
        for (int i = 0; i < n; i++)
        {
            push(str[i]);
        }
        string rvrsdStr = "";
        while (!isEmpty())
        {
            rvrsdStr += pop();
        }
        cout << "String D Reverse : " << rvrsdStr;
    }

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

```

```
    return 0;
}
```

Screenshoot program



```
PS D:\ITTP\Semester 2\Struktur data & Algoritma\Code\Praktikum\Modul 6> g++ unguided2.cpp -o unguided2
PS D:\ITTP\Semester 2\Struktur data & Algoritma\Code\Praktikum\Modul 6> .\unguided2
Kata : Jakarta keras
Kata di Reverse : sarek atrakaJ
PS D:\ITTP\Semester 2\Struktur data & Algoritma\Code\Praktikum\Modul 6>
```

Deskripsi program

Program ini menggunakan konsep stack untuk membalikkan string. Stack adalah struktur data LIFO (Last In, First Out) yang memungkinkan elemen yang terakhir masuk adalah yang pertama keluar. Dengan memanfaatkan stack, program ini dapat dengan mudah membalikkan urutan karakter dalam string. Input dari pengguna dibalik dan hasilnya ditampilkan di layar. Berikut penjelasan dari beberapa fungsi program diatas :

1. isFull()
Fungsi ini mengembalikan true jika stack penuh, yaitu ketika top sudah mencapai indeks maksimum (n-1).
2. isEmpty()
Fungsi ini mengembalikan true jika stack kosong, yaitu ketika top sama dengan -1.
3. push()
Fungsi ini menambahkan elemen x ke stack jika stack belum penuh. Elemen ditambahkan pada posisi ++top.
4. pop()
Fungsi ini mengeluarkan elemen teratas dari stack dan mengurangi nilai top satu satuan.
5. reverse()
Fungsi ini membalikkan urutan karakter dalam string str. Pertama, panjang string n diambil. Kemudian, setiap karakter dari string dimasukkan ke stack dengan memanggil fungsi push. Setelah semua karakter masuk ke stack, sebuah string kosong rvrsdStr disiapkan. Selama stack tidak kosong, karakter-karakter diambil satu per satu dari stack (menggunakan pop) dan ditambahkan ke rvrsdStr. Hasil akhir rvrsdStr adalah string yang sudah dibalik, dan dicetak ke layar.

BAB IV

KESIMPULAN

Stack adalah struktur data yang berfungsi sebagai tempat penyimpanan sementara dengan prinsip kerja LIFO (Last In, First Out), di mana elemen yang terakhir dimasukkan adalah yang pertama dikeluarkan. Ini mirip dengan tumpukan piring; piring yang terakhir diletakkan di atas akan diambil pertama. Kelebihan dari stack yaitu Sederhana dan efisien untuk masalah-masalah yang membutuhkan pemrosesan data dengan prinsip LIFO. Kekurangan dari stack adalah Terbatas pada akses elemen teratas saja; tidak cocok untuk aplikasi yang memerlukan akses acak ke elemen data.

DAFTAR PUSTAKA

Asisten Praktikum. (2024). MODUL 6 STACK. Learning Managament System

Adieb,Maulana. (04 Feb 2021). Memahami Stack, Struktur Data yang Mudahkan Programmer Jalankan Tugasnya. Diakses pada 21 Mei 2024 dari <https://glints.com/id/lowongan/stack-adalah/>