

LAPORAN PRAKTIKUM

Modul 9

TUMPUKAN



Disusun Oleh:

Adhiansyah Muhammad Pradana Farawowan - 2211104038

S1SE-07-02

Asisten Praktikum:

Aldi Putra

Andini Nur Hidayah

Dosen:

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

PROGRAM STUDI S1 REKAYASAN PERANGKAT LUNAK

FAKULTAS INFORMATIKA

UNIVERSITAS TELKOM PURWOKERTO

2024

A. Tujuan

Laporan praktikum ini memiliki tujuan di bawah berikut.

1. Memperkenalkan konsep *stack*, selanjutnya akan disebut *tumpukan*, sebagai salah satu struktur data
2. Memahami dan mengelola cara kerja tumpukan
3. Mengimplementasikan tumpukan dalam Bahasa C++

B. Landasan Teori

Tumpukan (*stack*) adalah struktur data yang bekerja dengan “data yang masuk terakhir adalah data sama yang akan keluar pertama” (*Last In, First Out*). Sebuah tumpukan memiliki variable *top* untuk melacak data yang berada paling atas. Tumpukan memiliki beberapa operasi di antaranya berikut. Sebuah tumpukan baru yang diinisialisasi memiliki *top* bernilai -1.

1. Mengecek isi penuh
Mengecek apakah isi tumpukan penuh dilakukan dengan operasi `is_full()`. Cara kerjanya adalah dengan mengecek apakah variabel *top* sama dengan batas tampung tumpukan.
2. Mengecek isi kosong
Mengecek apakah isi tumpukan kosong dilakukan dengan operasi `is_empty()`. Cara kerjanya adalah dengan mengecek apakah variabel *top* sama dengan -1.
3. Menambah isi
Menambah isi dilakukan dengan operasi `push()`. Dengan melakukan `push()`, ia akan juga menambah nilai *top* sebesar 1.
4. Mengurangi isi
Mengurangi isi dilakukan dengan operasi `pop()`. Dengan melakukan `pop()`, ia akan juga mengurangi nilai *top* sebesar 1.
5. Melihat data teratas
Melihat data teratas dilakukan dengan operasi `peek()`. Ini akan mencetak isi data yang berada di posisi teratas. Dengan kata lain, ia akan mencetak isi data yang berada di posisi *top*.
6. Mencetak data seluruh isi

C. Bimbingan (*guided*)

Pada bimbingan kali ini, kita akan mengimplementasikan tumpukan dengan larik dan daftar berantai. Kode akan dipaparkan di halaman selanjutnya.

guided_array.cxx

```
#define MAX 100
#include <iostream>

class Stack
{
private:
    int top;
    int arr[MAX];

public:
    Stack() { top = -1; };

    bool is_full()
    {
        return top == MAX - 1;
    }

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

    void push(int x)
    {
        if (is_full())
        {
            std::cout << "Stack penuh" << '\n';
            return;
        }

        arr[++top] = x;
    }

    void pop()
    {
        if (is_empty())
        {
            std::cout << "Stack kosong" << '\n';
            return;
        }

        top = top - 1;
    }

    int peek()
    {
        if (!is_empty())
        {
            return arr[top];
        }

        std::cout << "Stack kosong" << '\n';
        return -1;
    }

    void print_stack()
    {
        if (is_empty())
        {
            std::cout << "Stack kosong" << '\n';
            return;
        }

        for (int i = top; i >= 0; i = i - 1)
        {
            std::cout << arr[i] << " ";
        }
    }
}
```

```

    }

    std::cout << '\n';
}
};

int main()
{
    Stack tumpukanku;
    tumpukanku.push(10);
    tumpukanku.push(20);
    tumpukanku.push(30);

    std::cout << "Elemen-elemen tumpukan: ";
    tumpukanku.print_stack();

    std::cout << "Elemen teratas: " << tumpukanku.peek() << '\n';

    tumpukanku.pop();
    tumpukanku.pop();
    tumpukanku.pop();

    std::cout << "Elemen-elemen tumpukan setelah penghapusan: ";
    tumpukanku.print_stack();

    return 0;
}

```

Output dari guided_array.cxx

```

>a.exe
Elemen-elemen tumpukan: 30 20 10
Elemen teratas: 30
Elemen-elemen tumpukan setelah penghapusan: Stack kosong

```

guided_linked_list.cxx

```

#include <iostream>

class Node
{
public:
    int data;
    Node *next;
    Node(int data_value)
    {
        data = data_value;
        next = nullptr;
    }
};

class Stack
{
private:
    Node *top;

public:
    Stack()
    {
        top = nullptr;
    }

    bool is_empty()
    {
        return top == nullptr;
    }
}

```

```

void push(int x)
{
    Node *new_node = new Node(x);
    new_node->next = top;
    top = new_node;
}

void pop()
{
    if (is_empty())
    {
        std::cout << "Stack kosong" << '\n';
        return;
    }

    Node *temp = top;
    top = top->next;
    delete temp;
}

int peek()
{
    if (!is_empty())
    {
        return top->data;
    }

    std::cout << "Stack kosong" << '\n';
    return -1;
}

void print_stack()
{
    if (is_empty())
    {
        std::cout << "Stack kosong" << '\n';
        return;
    }

    Node *current = top;
    while (current)
    {
        std::cout << current->data << " ";
        current = current->next;
    }

    std::cout << '\n';
}

};

int main()
{
    Stack tumpukanku;
    tumpukanku.push(3);
    tumpukanku.push(5);
    tumpukanku.push(7);
    tumpukanku.push(11);
    tumpukanku.push(13);
    tumpukanku.push(17);

    std::cout << "Elemen-elemen tumpukan: ";
    tumpukanku.print_stack();

    std::cout << "Elemen teratas: " << tumpukanku.peek() << '\n';
}

```

```

tumpukanku.pop();
tumpukanku.pop();
tumpukanku.pop();

std::cout << "Elemen-elemen tumpukan setelah penghapusan: ";
tumpukanku.print_stack();

return 0;
}

```

Output dari guided_linkedlist.cxx

```

>a.exe
Elemen-elemen tumpukan: 17 13 11 7 5 3
Elemen teratas: 17
Elemen-elemen tumpukan setelah penghapusan: 7 5 3

```

D. Tugas Mandiri (*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

unguided_1.cpp

```

#define MAX 100
#include <iostream>
#include <string>
#include <cctype>

class Stack
{
private:
    int top;
    std::string arr[MAX];

public:
    Stack() { top = -1; };

    bool is_full()
    {
        return top == MAX - 1;
    }

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

    void push(std::string x)
    {
        if (is_full())
        {
            std::cout << "Stack penuh" << '\n';
            return;
        }

        arr[++top] = x;
    }

    void pop()
    {

```

```

        if (is_empty())
        {
            std::cout << "Stack kosong" << '\n';
            return;
        }

        top = top - 1;
    }

    std::string peek()
    {
        if (!is_empty())
        {
            return arr[top];
        }

        std::cout << "Stack kosong" << '\n';
        return "";
    }

    std::string at_index(int i) {
        return arr[i];
    }

    void print_stack()
    {
        if (is_empty())
        {
            std::cout << "Stack kosong" << '\n';
            return;
        }

        for (int i = top; i >= 0; i = i - 1)
        {
            std::cout << arr[i] << " ";
        }

        std::cout << '\n';
    }
};

int main() {
    std::string masukan_pengguna;
    Stack data_asli;
    Stack data_terbaik;

    std::cout << "Masukkan data untuk dicek palindromnya: ";
    std::getline(std::cin, masukan_pengguna);

    for (int i = 0; i < masukan_pengguna.length(); i = i + 1) {
        char dari_awal = std::tolower(masukan_pengguna[i]);
        char dari_akhir =
std::tolower(masukan_pengguna[(masukan_pengguna.length() - 1) - i]);

        if (std::isspace(dari_awal) != 1) {
            data_asli.push(std::string(1, dari_awal));
        }

        if (std::isspace(dari_akhir) != 1) {
            data_terbaik.push(std::string(1, dari_akhir));
        }

        if (data_terbaik.peek() != data_asli.peek()) {
            std::cout << "Bukan palindrom" << '\n';
            return 0;
        }
    }
}

```

```

std::cout << "Betul ini palindrom" << '\n';
return 0;
}

```

Pada dasarnya, program ini bekerja dengan memasukkan tiap karakter masukan ke dalam tumpukan yang berisi string normal dan string yang terbaik, tetapi sesudah tiap karakter yang masuk dibandingkan.

```

>a.exe
Masukkan data untuk dicek palindromnya: tanpa kamu menemaniku
Bukan palindrom

>a.exe
Masukkan data untuk dicek palindromnya: kasur ini rusak
Betul ini palindrom

```

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?

```

unguided_2.cpp
#define MAX 100
#include <iostream>
#include <string>

class Stack
{
private:
    int top;
    std::string arr[MAX];

public:
    Stack() { top = -1; };

    bool is_full()
    {
        return top == MAX - 1;
    }

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

    void push(std::string x)
    {
        if (is_full())
        {
            std::cout << "Stack penuh" << '\n';
            return;
        }

        arr[++top] = x;
    }

    void pop()
    {
        if (is_empty())
        {
            std::cout << "Stack kosong" << '\n';
            return;
        }
    }
}

```



```

        top = top - 1;
    }

    std::string peek()
    {
        if (!is_empty())
        {
            return arr[top];
        }

        std::cout << "Stack kosong" << '\n';
        return "";
    }

    void print_stack()
    {
        if (is_empty())
        {
            std::cout << "Stack kosong" << '\n';
            return;
        }

        for (int i = top; i >= 0; i = i - 1)
        {
            std::cout << arr[i];
        }

        std::cout << '\n';
    }
};

int main() {
    std::string masukan_pengguna;
    std::cout << "Masukkan data untuk dibalik: ";
    std::getline(std::cin, masukan_pengguna);

    Stack yang_terbalik;

    for (int i = 0; i < masukan_pengguna.length(); i = i + 1) {
        char dari_akhir = masukan_pengguna[i];
        yang_terbalik.push(std::string(1, dari_akhir));
    }

    std::cout << "Hasil: ";
    yang_terbalik.print_stack();
}

```

Dengan menggunakan sifat print_stack() yang mencetak terbalik, program ini bekerja dengan menggunakan sifat dari Stack itu sendiri.

```

>a.exe
Masukkan data untuk dibalik: Hidup Untukmu, Mati Tanpamu
Hasil: umapnaT itaM ,umkutnU pudih

```