

LAPORAN PRAKTIKUM

MODUL VI STACK



Disusun oleh:
Geranada Saputra Priambudi
NIM: 2311102008

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

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM 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

Struktur Data Stack adalah jenis struktur data yang beroperasi berdasarkan prinsip Last In First Out (LIFO). Elemen terbaru yang ditambahkan ke stack akan menjadi elemen yang pertama dikeluarkan.

Operasi yang umum dilakukan pada stack adalah push (menambah elemen baru), pop (menghapus elemen teratas), dan peek (melihat elemen teratas tanpa menghapusnya). Stack sering digunakan dalam pemrograman, seperti pengelolaan memori, pengecekan tata bahasa, dan algoritma backtracking.

Jenis Struktur Data Stack

Ada dua jenis utama dari struktur data stack: stack statis dan stack dinamis.

1. Stack Statis

Stack ini didefinisikan dengan ukuran yang tetap dan tidak dapat diubah selama proses eksekusi. Elemen dapat ditambahkan atau dihapus dari stack, tetapi ukurannya tetap sama. Stack statis dapat diimplementasikan dengan menggunakan array atau pointer.

2. Stack Dinamis

Stack ini didefinisikan dengan ukuran yang dapat diubah sesuai dengan jumlah elemen yang ada di dalamnya. Saat jumlah elemen di dalam stack melebihi kapasitas yang telah ditentukan, kapasitas stack akan ditambahkan secara otomatis. Stack dinamis dapat diimplementasikan dengan menggunakan linked list atau array dinamis.

Selain itu, stack juga dapat diimplementasikan menggunakan berbagai metode seperti array, linked list, dan array dinamis.

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 D:\Telkom University\Teknik Informatika\Semester 2\Praktikum Struktur data dan Algoritma\Module6> cd "d:\Telkom University\Teknik Informatika\Semester 2\Praktikum
Struktur data dan Algoritma\Module6\" ; if ($?) { g++ Guided.cpp -o Guided } ; if ($?) { .\Guided }
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 D:\Telkom University\Teknik Informatika\Semester 2\Praktikum Struktur data dan Algoritma\Module6>

```

Deskripsi program

Program di atas adalah implementasi dari sebuah stack menggunakan array dalam bahasa pemrograman C++. Stack adalah struktur data yang mengikuti prinsip LIFO (Last In, First Out), di mana elemen terakhir yang dimasukkan adalah elemen pertama yang akan dikeluarkan.

LATIHAN KELAS - UNGUIDED

1. Unguided 1

Source code

```
#include <iostream>
#include <stack>
#include <string>

using namespace std;

// Menghapus spasi dan karakter non-alfabet dari string
string cleanString(const string& str) {
    string cleanedStr;
    for (char c : str) {
        if (isalpha(c)) {
            cleanedStr += tolower(c);
        }
    }
    return cleanedStr;
}

// Memeriksa apakah sebuah kalimat adalah palindrom
bool isPalindrome(const string& str) {
    stack<char> charStack;
    string cleanedStr = cleanString(str);

    // Memasukkan setengah pertama dari kalimat ke dalam stack
    int length = cleanedStr.length();
    for (int i = 0; i < length / 2; ++i) {
        charStack.push(cleanedStr[i]);
    }

    // Membandingkan setengah kedua dari kalimat dengan stack
```

```

        int i = (length + 1) / 2;
        while (i < length) {
            if (cleanedStr[i] != charStack.top()) {
                return false;
            }
            charStack.pop();
            ++i;
        }

        return true;
    }

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

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

        return 0;
    }
}

```

Screenshoot program

```

PS D:\Telkom University\Teknik Informatika\Semester 2\Praktikum Struktur data dan Algoritma\Module> cd "d:\Telkom University\Teknik Informatika\Semester 2\Praktikum
Struktur data dan Algoritma\Module6\" ; if ($?) { g++ Unguided_1.cpp -o Unguided_1 } ; if ($?) { .\Unguided_1 }
Masukkan kalimat: apa
Kalimat tersebut adalah palindrom.
PS D:\Telkom University\Teknik Informatika\Semester 2\Praktikum Struktur data dan Algoritma\Module6> cd "d:\Telkom University\Teknik Informatika\Semester 2\Praktikum
Struktur data dan Algoritma\Module6\" ; if ($?) { g++ Unguided_1.cpp -o Unguided_1 } ; if ($?) { .\Unguided_1 }
Masukkan kalimat: Telkom University
Kalimat tersebut bukan palindrom.
PS D:\Telkom University\Teknik Informatika\Semester 2\Praktikum Struktur data dan Algoritma\Module6>

```

Deskripsi program

Program di atas adalah implementasi untuk memeriksa apakah sebuah kalimat adalah palindrom atau tidak, menggunakan struktur data stack dalam bahasa pemrograman C++. Sebuah palindrom adalah kata, frasa, angka, atau urutan karakter lainnya yang sama jika dibaca mundur sekalipun (dengan mengabaikan spasi, tanda baca, dan perbedaan huruf besar-kecil).

2. Unguided 2

Source code

```
#include <iostream>
#include <stack>
#include <string>

using namespace std;

int main() {
    string kalimat;
    stack<char> s;

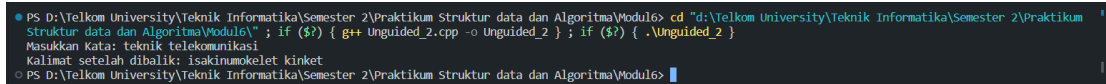
    cout << "Masukkan Kata: ";
    getline(cin, kalimat);

    // Push characters of the sentence into the stack
    for (char c : kalimat) {
        s.push(c);
    }

    // Pop characters from the stack and print them
    cout << "Kalimat setelah dibalik: ";
    while (!s.empty()) {
        cout << s.top();
    }
}
```

```
s.pop();  
}  
  
cout << endl;  
  
return 0;  
}
```

Screenshot program



```
PS D:\Telkom University\Teknik Informatika\Semester 2\Praktikum Struktur data dan Algoritma\Module6> cd "d:\Telkom University\Teknik Informatika\Semester 2\Praktikum Struktur data dan Algoritma\Module6\" ; if ($?) { g++ Unguided_2.cpp -o Unguided_2 } ; if ($?) { .\Unguided_2 }  
Masukkan Kata: teknik telekomunikasi  
Kalimat setelah dibalik: isakinumokelet kinket  
PS D:\Telkom University\Teknik Informatika\Semester 2\Praktikum Struktur data dan Algoritma\Module6>
```

Deskripsi program

Program di atas adalah implementasi sederhana yang membalik sebuah kalimat yang dimasukkan oleh pengguna menggunakan struktur data stack dalam bahasa pemrograman C++.

BAB IV

KESIMPULAN

Praktikum ini bertujuan untuk memahami konsep dasar, implementasi, dan aplikasi dari struktur data stack. Stack adalah salah satu struktur data fundamental yang mengikuti prinsip LIFO (Last In, First Out), di mana elemen yang terakhir dimasukkan adalah elemen pertama yang akan dikeluarkan.

Stack adalah struktur data yang sangat berguna untuk banyak aplikasi yang membutuhkan pengolahan data dengan urutan LIFO. Operasi dasar seperti push dan pop sangat intuitif dan mudah diimplementasikan. Stack dapat digunakan untuk memecahkan berbagai masalah, seperti pemeriksaan palindrom dan pembalikan urutan karakter. Struktur data ini sering digunakan dalam algoritma seperti pemrosesan ekspresi matematika, penelusuran graf, dan pelacakan riwayat (seperti dalam browser web). Operasi stack biasanya memiliki kompleksitas waktu $O(1)$, membuatnya sangat cepat untuk digunakan dalam aplikasi yang memerlukan penyimpanan dan pengambilan data sementara.