

LAPORAN PRAKTIKUM

MODUL VI “STACK”



Disusun oleh:
Andika Indra Prastawa
NIM: 2311102033

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

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

BAB I

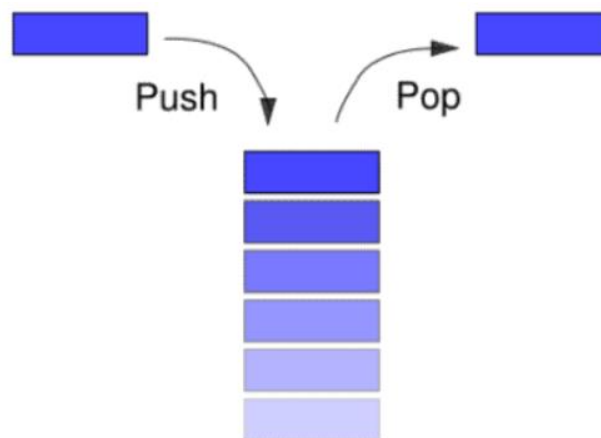
TUJUAN PRAKTIKUM

- 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

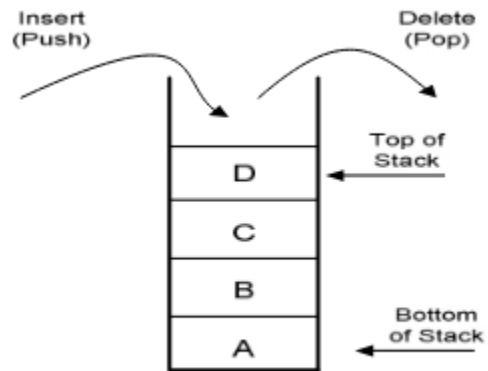
Stack adalah sebuah kumpulan data dimana data yang diletakkan di atas data yang lain. Dengan demikian stack adalah struktur data yang menggunakan konsep LIFO (Last In First Out). Dengan demikian, elemen terakhir yang disimpan dalam stack menjadi elemen pertama yang diambil. Dalam proses komputasi, untuk meletakkan sebuah elemen pada bagian atas dari stack, maka dilakukan operasi push. Dan untuk memindahkan dari tempat yang atas tersebut, maka dilakukan operasi pop



stack, maka dilakukan operasi **push**. Dan untuk memindahkan dari tempat yang atas tersebut, maka dilakukan operasi **pop**. Berikut ini adalah operator-operator atau nama method yang biasa digunakan dalam pemrograman algoritma stack.

- **PUSH**: penyisipan (Memasukkan elemen).
- **POP**: penghapusan (Mengeluarkan elemen puncak).
- **IsEmpty**: operator yang memeriksa apakah stack kosong.

- ***IsFull***: operator yang memeriksa apakah stack penuh.
- ***Clear***: operator untuk menghapus semua elemen didalam stack.



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).

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:\praktikum_1\praktikum_1> cd "c:\praktikum_1\praktikum_1\" ;  
6_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
```

Deskripsi program

Program ini mengimplementasikan struktur data stack menggunakan array dengan kapasitas 5 elemen, yang bekerja berdasarkan prinsip Last-In, First-Out (LIFO). Program ini memiliki beberapa fungsi: untuk menambahkan elemen (push), menghapus elemen teratas (pop), melihat elemen di posisi tertentu (peek), menghitung jumlah elemen dalam stack, serta mengubah dan mengosongkan stack. Selain itu, program ini juga memeriksa apakah stack dalam keadaan penuh atau kosong. Misalnya, program dapat menambahkan lima elemen, mencetaknya, menghapus satu elemen, mengganti elemen di posisi tertentu, dan akhirnya mengosongkan seluruh stack. Fungsi-fungsi ini menunjukkan operasi dasar yang dapat dilakukan pada stack.

LATIHAN KELAS – UNGUIDED

1. Unguided 1

Source code

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

bool SystemPalindrom2311102033(const string& input) {
    stack<char> charCharacters;
    for (char c : input) {
        charCharacters.push(c);
    }

    for (char c : input) {
        if (c!= charCharacters.top()) {
            return false;
        }
        charCharacters.pop();
    }

    return true;
}

int main() {
    string input;
    cout << "Masukan Kalimat: ";
    getline(cin, input);
}
```

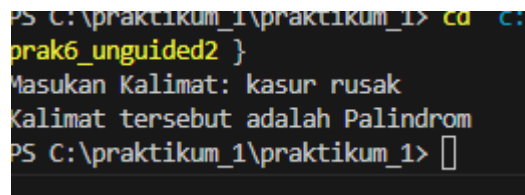
```

    if (SystemPalindrom2311102033(input)) {
        cout << "Kalimat tersebut adalah Palindrom" << endl;
    } else {
        cout << "Kalimat tersebut bukan Palindrom" << endl;
    }

    return 0;
}

```

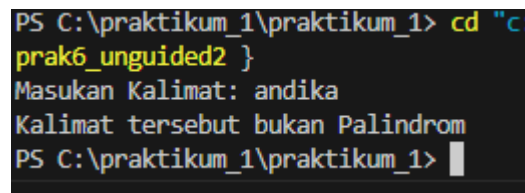
Screenshoot program



```

PS C:\praktikum_1\praktikum_1> cd "c:\praktikum_1\praktikum_1"
prak6_unguided2 }
Masukan Kalimat: kasur rusak
Kalimat tersebut adalah Palindrom
PS C:\praktikum_1\praktikum_1>

```



```

PS C:\praktikum_1\praktikum_1> cd "c:\praktikum_1\praktikum_1"
prak6_unguided2 }
Masukan Kalimat: andika
Kalimat tersebut bukan Palindrom
PS C:\praktikum_1\praktikum_1>

```

Deskripsi program

Program ini memeriksa apakah sebuah kalimat adalah palindrome, yaitu kalimat yang dapat dibaca sama dari depan dan belakang. Program menggunakan stack untuk menyimpan karakter-karakter dari input dan kemudian membandingkannya dengan urutan asli untuk melihat apakah mereka sama. Jika semua karakter cocok, kalimat dianggap palindrome; jika tidak, kalimat bukan palindrome. Program meminta pengguna untuk memasukkan sebuah kalimat dan kemudian memberikan hasil apakah kalimat tersebut adalah palindrome. Namun, program ini tidak mengabaikan spasi dan variasi kasus, jadi hasilnya mungkin kurang akurat jika ada spasi atau perbedaan kapitalisasi dalam kalimat. Untuk pemeriksaan yang lebih akurat, diperlukan praproses untuk menghilangkan spasi dan menyamakan kasus huruf sebelum mengecek palindrome.

2. Unguided 2

Source code

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

using namespace std;

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

    stack<char> s2311102033;
    for (char c : input) {
        s2311102033.push(c);
    }

    string reversed;
    while (!s2311102033.empty()) {
        char c = s2311102033.top();
        s2311102033.pop();
        reversed += c;
    }

    size_t start = 0;
    size_t end = 0;
    for (size_t i = 0; i < reversed.length(); i++) {
        if (reversed[i] == ' ') {
            end = i;
            string word = reversed.substr(start, end - start);
```

```

        cout << word << ' ';
        start = i + 1;
    }

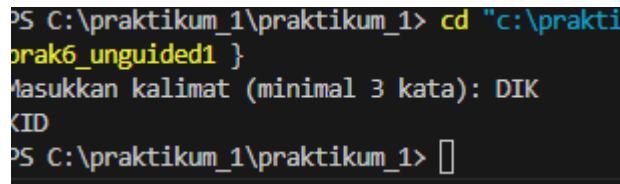
}

string lastWord = reversed.substr(start);
cout << lastWord << endl;

return 0;
}

```

Screenshoot program



```

PS C:\praktikum_1\praktikum_1> cd "c:\praktikum_1\praktikum_1"
PS C:\praktikum_1\praktikum_1> g++ prak6_unguided1.cpp
Masukkan kalimat (minimal 3 kata): DIK
KID
PS C:\praktikum_1\praktikum_1>

```

Deskripsi program

Program di atas menerima input sebuah kalimat yang terdiri dari minimal tiga kata dari pengguna. Kemudian, menggunakan struktur data stack, program membalik urutan karakter dari kalimat tersebut. Setelah itu, program memisahkan kata-kata dari hasil pembalikan tersebut dan mencetaknya dalam urutan terbalik. Program ini memberikan penggunaan praktis dari stack dalam memanipulasi urutan data dan memisahkan komponen-komponen dalam sebuah teks.

BAB IV

KESIMPULAN

Dalam makalah ini, kita telah membahas konsep dasar stack dalam pemrograman C++, termasuk struktur, operasi, dan implementasinya. Stack adalah struktur data yang mengikuti prinsip LIFO (Last In, First Out), di mana elemen terakhir yang masuk adalah elemen pertama yang keluar. Ini memiliki beragam aplikasi dalam pemrograman, seperti pengelolaan fungsi rekursif, evaluasi ekspresi matematika, dan implementasi undo/redo dalam aplikasi. Operasi dasar stack meliputi push, pop, top, dan isEmpty, yang memungkinkan penambahan, penghapusan, dan pemeriksaan elemen di stack. Dalam C++, stack dapat diimplementasikan menggunakan array atau daftar tertaut (linked list), dengan perpustakaan standar C++ (STL) menyediakan kelas stack yang mempermudah penggunaan stack dengan metode-metode yang sudah terdefinisi.

DAFTAR PUSTAKA

- [1] Karumanchi, N. (2016). Data Structures and algorithms made easy: Concepts, problems, Interview Questions. CareerMonk Publications.
- [2] https://www.academia.edu/30019746/C_Stack_pdf
- [3] <https://sisfo.itp.ac.id/bahanajar/BahanAjar/Anisya/Modul%203%20-%20Tumpukan.pdf>