

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL 6
“STACK”**



**DISUSUN OLEH :
RIZKY PERLINTA SEMBIRNG
2311102061**

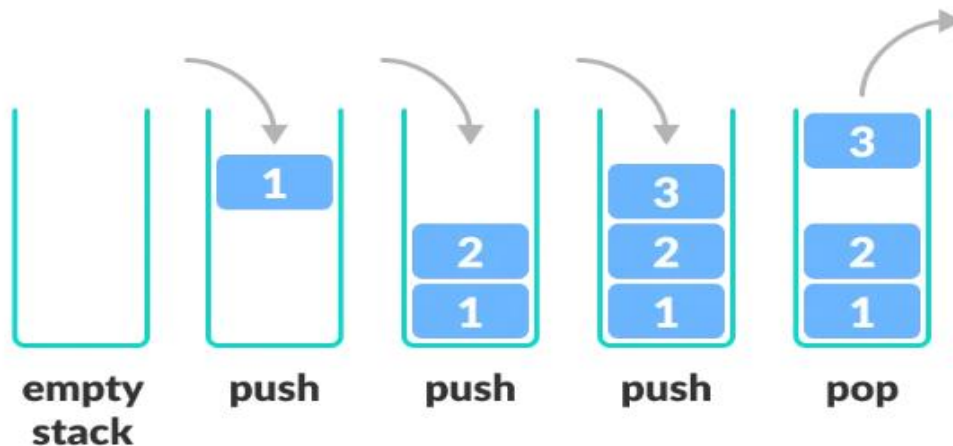
**DOSEN
WAHYU ANDI SAPUTRA, S.PD., M.PD.**

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

A. DASAR TEORI

1. STACK

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.



2. Karakteristik STACK

Struktur data stack memiliki ciri sebagai berikut:

- Stack digunakan pada banyak algoritma yang berbeda seperti Tower of Hanoi, Tree traversal, rekursi dll.
- Stack diimplementasikan dengan struktur data array atau linked list.
- Mengikuti prinsip operasi Last In First Out, yaitu elemen yang dimasukkan pertama akan muncul terakhir dan sebaliknya.
- Penyisipan dan penghapusan terjadi di satu ujung yaitu dari atas tumpukan.
- Apabila ruang memori yang dialokasikan untuk struktur data stack sudah penuh namun masih dilakukan operasi penyisipan elemen maka akan terjadi stack overflow.
- Apabila struktur data tidak memiliki elemen data atau kosong, namun tetap dilakukan operasi penghapusan maka akan terjadi stack underflow.

3. Operasi – Operasi Dasar Pada Stack

Ada beberapa operasi dasar yang bisa kita lakukan terhadap struktur data stack.

Operasi-operasi tersebut meliputi:

- **Push:** Menyisipkan elemen ke bagian atas stack
- **Pop:** Menghapus elemen atas dari stack
- **IsEmpty:** Memeriksa apakah stack kosong
- **IsFull:** Memeriksa apakah stack sudah penuh
- **Peek:** Mendapatkan nilai elemen teratas tanpa menghapusnya

4. Fungsi Dan Kegunaan Stack

Adapun fungsi dan kegunaan struktur data stack adalah sebagai berikut:

- Struktur data stack digunakan dalam evaluasi dan konversi ekspresi aritmatika, Proses ini banyak dipakai untuk program kompiler.
- Stack digunakan dalam pemrograman rekursi.
- Digunakan untuk pemeriksaan tanda kurung.
- Stack digunakan dalam manajemen memori.
- Dipakai untuk memproses pemanggilan sebuah fungsi.

Salah satu contoh penerapan struktur data stack adalah fitur tombol back pada browser. Dimana browser akan menyimpan semua URL yang telah kita kunjungi sebelumnya dalam stack. Setiap kali kita mengunjungi halaman baru, halaman itu ditambahkan di atas stack. Saat kita menekan tombol kembali, URL saat ini dihapus dari tumpukan, dan URL sebelumnya diakses.

B. Guided

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;
}

```

Screenshot Output

The screenshot shows the output of a C++ program in a dark-themed terminal. The output lists the books in the stack: Inggris, Dasar Multimedia, Matematika Diskrit, Struktur Data, and Kalkulus. It then checks if the stack is full (1) and empty (0). It prints the element at index 2 (Dasar Multimedia) and the number of elements (4). After changing the element at index 2 to Bahasa Jerman and printing the stack again, it destroys the stack. Finally, it prints the number of elements after deletion (0) and states that no data was printed.

Overlaid on the bottom right is a Notepad window with the following text:

```

Nama : Rizky Perlinta Sembiring
Nim : 2311102061
Kelas : IF-11

```

The Notepad window's status bar shows: Ln 3, Col 14 | 65 characters | 100% | Window | UTF-8.

Deskripsi Program:

C. 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.

Contoh:

Kalimat : ini

Kalimat tersebut adalah polindrom

Kalimat : telkom

Kalimat tersebut adalah bukan polindrom

Source Code:

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

using namespace std;

bool isPalindrome(const string &str) {
    stack<char> s;
    string cleanedStr;

    for (char ch : str) {
        if (isalpha(ch)) {
            cleanedStr += tolower(ch);
        }
    }

    for (char ch : cleanedStr) {
        s.push(ch);
    }

    for (char ch : cleanedStr) {
        if (ch != s.top()) {
            return false;
        }
        s.pop();
    }
}
```



```

    }

    return true;
}

int main() {
    string input;

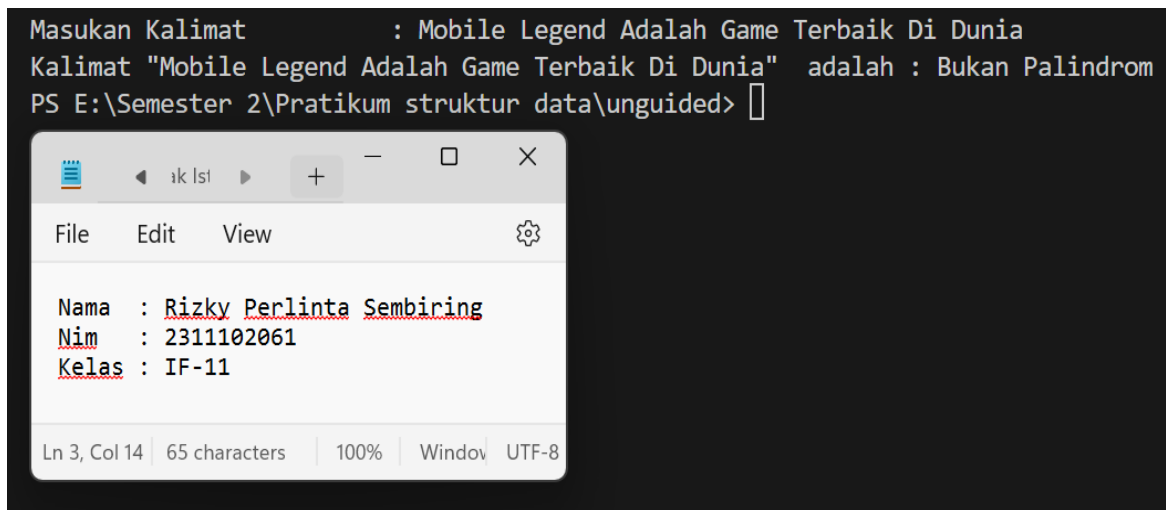
    cout << "Masukan Kalimat\t\t : ";
    getline(cin, input);

    if (isPalindrome(input)) {
        cout << "Kalimat \"" << input << "\" adalah\t : Palindrom" <<
endl;
    } else {
        cout << "Kalimat \"" << input << "\" adalah : Bukan Palindrom"
<<
endl;
    }

    return 0;
}

```

ScreenShot Output:



The screenshot shows the output of a C++ program in a terminal window. The program prompts the user to enter a sentence. The user enters "Mobile Legend Adalah Game Terbaik Di Dunia". The program outputs "Kalimat "Mobile Legend Adalah Game Terbaik Di Dunia" adalah : Bukan Palindrom". Below the terminal output, there is a Notepad++ window showing a text file with the following content:

```

Nama : Rizky Perlinta Sembiring
Nim : 2311102061
Kelas : IF-11

```

The Notepad++ window also shows the status bar at the bottom: "Ln 3, Col 14 | 65 characters | 100% | Window | UTF-8".

Deskripsi Program: Program ini merupakan sebuah alat sederhana untuk mengecek apakah sebuah kalimat adalah palindrom. Dengan menerima input kalimat dari pengguna, program membersihkan kalimat tersebut dari karakter-karakter non-alfabet dan mengubahnya menjadi huruf kecil. Kemudian, ia membandingkan setiap karakter dari kalimat dengan karakter-karakter dalam stack, dimulai dari belakang. Jika tidak ada perbedaan, program menyatakan bahwa kalimat tersebut adalah palindrom; jika tidak,

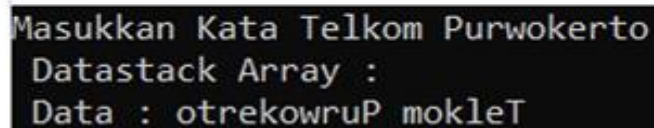
kalimat tersebut bukan palindrom. Selanjutnya, hasil pengecekan ditampilkan kepada pengguna.

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?

Contoh

Kalimat : Telkom Purwokerto

Hasil : otrekowruP mokleT



```
Masukkan Kata Telkom Purwokerto
Datastack Array :
Data : otrekowruP mokleT
```

Source Code:

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

using namespace std;

string reverseWords(const string &str) {
    stack<char> s;
    string result = "";

    //Membalikkan setiap karakter dalam string menggunakan stack
    for (char ch : str) {
        s.push(ch);
    }

    //Mengambil karakter dari stack untuk membentuk kalimat yang dibalik
    while (!s.empty()) {
        result += s.top();
        s.pop();
    }

    return result;
}

int main() {
    string input;
```

```

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

//Memastikan bahwa input memiliki minimal 3 kata
int wordCount = 0;
for (char ch : input) {
    if (ch == ' ') {
        wordCount++;
    }
}
wordCount++; //Menambah satu untuk kata terakhir

if (wordCount < 3) {
    cout << "Error: Kalimat harus memiliki minimal 3 kata." << endl;
} else {
    string reversed = reverseWords(input);
    cout << "Hasil: " << reversed << endl;
}

return 0;
}

```

ScreenShot Output:

```

Masukkan kalimat (minimal 3 kata): Air Tenang Menghanyutkan
Hasil: naktuynahgneM gnaneT riA
PS E:\Semester 2\Pratikum struktur data\unguided>

```

Notepad++ window content:

```

Nama : Rizky Perlinta Sembiring
Nim : 2311102061
Kelas : IF-11

```

Notepad++ status bar: Ln 3, Col 14 | 65 characters | 100% | Window | UTF-8

Deskripsikan Program: Program diatas adalah program mengembalikan urutan kata-kata. pertama user akan disuruh menginputkan 3 kata. setelah itu program akan memeriksa apakah user telah menginputkan 3 kata, jika tidak maka program akan langsung ERROR DAN jika user memasukkan 3 kata maka program akan mengoutputkan kata terbalik dari kata yang telah diinputkan oleh user.

D. Kesimpulan

Pada “STACK” Data hanya bisa di ambil secara berurutan, tidak bias di ambil secara langsung. Hal ini berada dengan system yang ada pada “LINKED LIST” di mana data dapat di ambil secara acak, bias di tengah, di atas atau terakhir, bahkan data dapat langsung di hapus.

E. Referensi

[1] Asisten Pratikum “Modul 4 HASH TABLE”, Learning Management System, 2024.

[2] Trivusi, “Struktur Data Stack: Pengertian, Karakteristik, dan Kegunaannya”, (2022), diakses pada 20 Mei 2024, dari

<https://www.trivusi.web.id/2022/07/struktur-data-stack.html>

[3] N, Sigit, “Contoh Membuat Program dengan Konsep Stack C++”, (2023), diakses pada 20 mei 2024, dari

<https://pemburukode.com/contoh-program-konsep-stack-cpp/>