

**LAPORAN PRAKTIKUM
STRUKTUR DATA DAN ALGORITMA**

**MODUL VI
STACK**



DISUSUN OLEH :
MUHAMAD NASRULLOH
2311102044

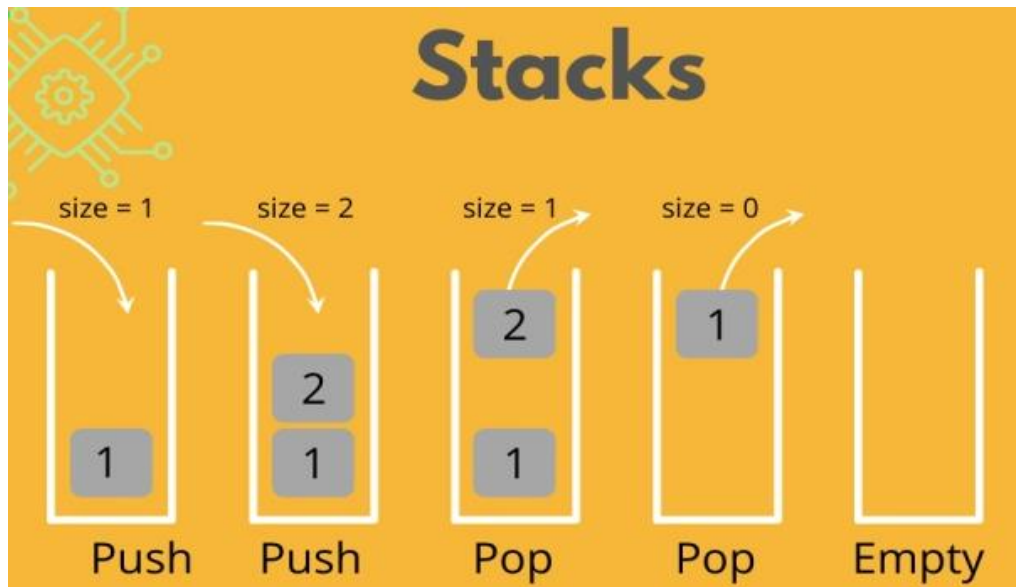
DOSEN :
WAHYU ANDI SAPUTRA, S.PD., M.ENG.

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

A. DASAR TEORI

- Pengertian

Stack dalam C++ adalah struktur data yang mengikuti prinsip Last In First Out (LIFO), yang berarti elemen terakhir yang dimasukkan ke dalam stack akan pertama kali diambil. Jadi bisa dikatakan stack adalah tumpukan (dalam Bahasa Indonesia) yang bila divisualisasikan ibarat wadah dimana kita dapat menyimpan elemen dan menghapusnya.



Dari ilustrasi gambar diatas diketahui setiap kita ingin menambahkan elemen ke tumpukan itu ditambahkan di bagian atas dan setiap kali ingin menghapus elemen dari tumpukan itu juga dimulai dari bagian atasnya.

- Bentuk Operasi atau Fungsi

1. Push : menyisipkan atau menambahkan elemen baru ke bagian atas stack
2. Pop : menghapus satu elemen atas dari stack
3. IsEmpty : memeriksa apakah stack kondisi kosong
4. IsFull : memeriksa apakah stack sudah penuh
5. Peek/Top : mendapatkan (mengetahui) nilai elemen teratas tanpa perlu menghapusnya.

- Jenis – Jenis Stack

1. Register Stack : jenis stack yang hanya mampu menampung data dalam jumlah kecil, hal ini karena ukuran unit memorinya sangat kecil dibandingkan memori stack.

2. Memory Stack : dengan kedalaman stack yang cukup fleksibel dan mampu menangani dalam skala yang lebih besar dibanding jenis sebelumnya.

- **Kegunaan Stack**

Stack digunakan dalam evaluasi dan konversi ekspresi aritmatika yang banyak dipakai untuk program compiler; digunakan dalam pemrograman rekursi; untuk pemeriksaan tanda kurung; digunakan dalam manajemen memori; dan untuk memproses pemanggilan sebuah fungsi.

- **Kekurangan Stack**

Diantaranya ukuran memori yang terbatas, kemungkinan terjadi stack overflow (dimana ruang memori yang dialokasikan untuk stack sudah penuh namun masih dilakukan operasi penyisipan elemen/push), tidak dapat mengakses data/elemen secara acak (karena data yang dapat diakses adalah data yang teratas), dan dapat menyebabkan fungsi tak terdefinisi (ketika penyimpanan variabel ditimpa).

B. GUIDED

- Guided I

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()
{

```

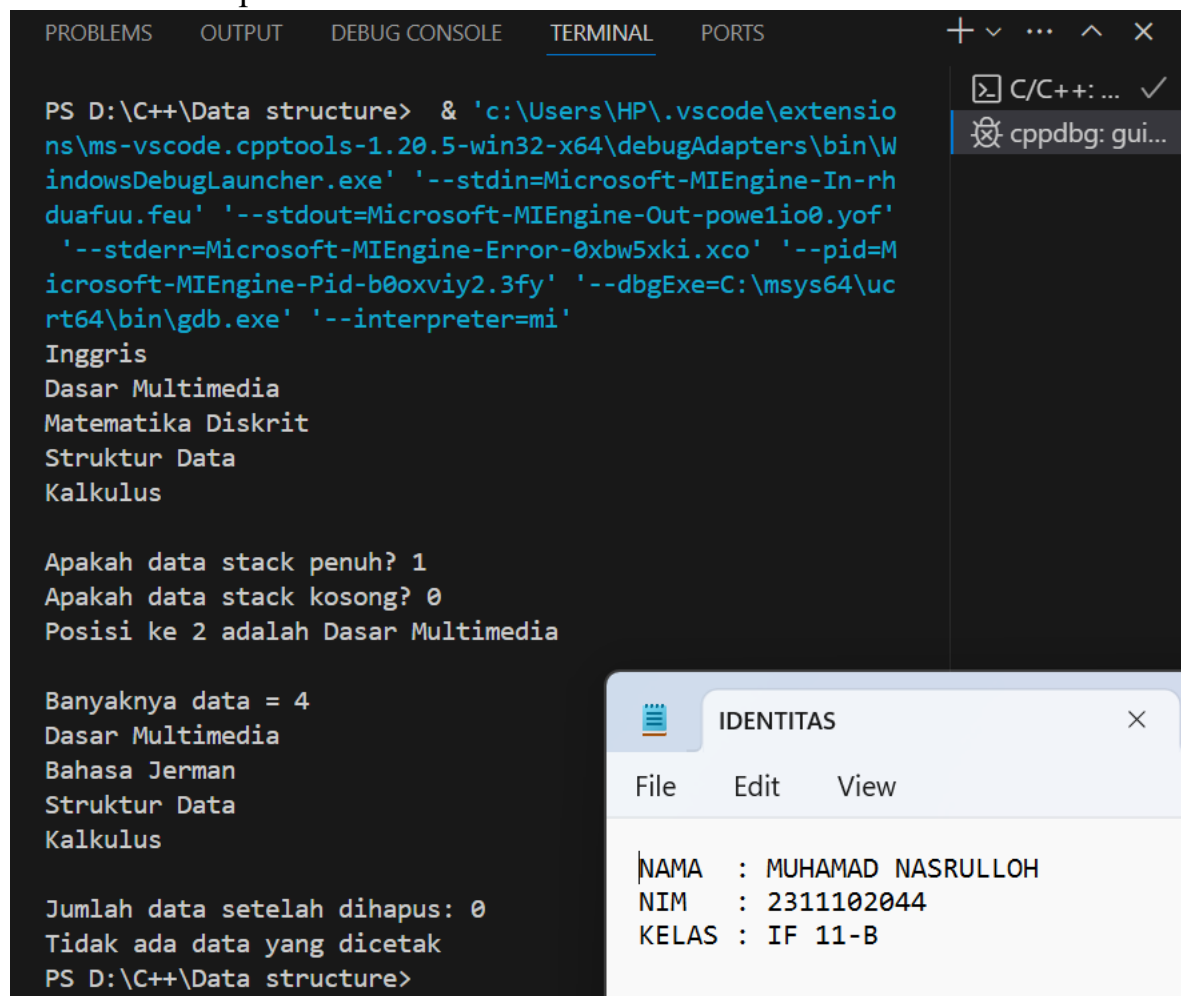
```

    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 a VS Code interface with a terminal window. The terminal output displays the execution of a C++ program that manages a stack of subjects. The output includes the list of subjects, checks for a full/empty stack, and the final count of elements. A small 'IDENTITAS' window is also visible in the foreground.

```
PS D:\C++\Data structure> & 'c:\Users\HP\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-rhduafoo.feu' '--stdout=Microsoft-MIEngine-Out-powellio0.yof' '--stderr=Microsoft-MIEngine-Error-0xbw5xki.xco' '--pid=Microsoft-MIEngine-Pid-b0oxviy2.3fy' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
```

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:\C++\Data structure>

IDENTITAS

	File	Edit	View
NAMA	MUHAMAD NASRULLOH		
NIM	2311102044		
KELAS	IF 11-B		

Deskripsi

Program diatas adalah implementasi stack pada c++. Dengan operasi dasar yang dilakukan seperti push, pop, peek dan manipulasi lainnya menggunakan array statis. Fungsi tersebut memastikan operasi itu dilakukan dengan batasan stack (apakah penuh atau kosong). Deklarasi : `arrayBuku[5]` : array untuk menyimpan data stack maksimal 5 elemen. Fungsi : `isFull()` untuk mengecek apakah stack sudah penuh, `isEmpty()` untuk mengecek apakah stack kosong, `pushArrayBuku(string data)` untuk menambahkan data ke stack jika belum penuh, `popArrayBuku()` untuk menghapus data stack teratas jika tidak kosong, `peekArrayBuku(int posisi)` untuk melihat data pada posisi tertentu dalam stack, `countStack()` untuk mengembalikan jumlah data dalam stack. Dst.

C. UNGUIDED

- Unguided I

Buatlah program untuk menentukan apakah kalimat tersebut yang di inputkan dalam program stack adalah palindrom atau tidak. Palindrom kalimat yang dibaca dari depan dan belakang sama. Jelaskan bagaimana cara kerja programnya.

Source Code

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

using namespace std;

bool isPalindrome(string str) {
    // Konversi semua huruf menjadi huruf kecil
    for (int i = 0; i < str.length(); i++) {
        str[i] = tolower(str[i]);
    }

    // Buat stack untuk menampung karakter kalimat
    stack<char> s;

    // Masukkan karakter kalimat ke stack
    for (int i = 0; i < str.length(); i++) {
        s.push(str[i]);
    }

    // Bandingkan karakter dari depan dan belakang kalimat
    for (int i = 0; i < str.length() / 2; i++) {
        if (str[i] != s.top()) {
            return false;
        }
        s.pop();
    }

    return true;
}

int main() {
    string input;
    int pilihan;

    do {
        cout << "\nMenu Program Pemeriksa Kalimat Palindrom:\n";
        cout << "1. Masukkan Kata/Kalimat\n";
        cout << "2. Exit\n";
```



```

    cout << "Pilihan Anda: ";
    cin >> pilihan;

    if (pilihan == 1) {
        cout << "Masukkan kata/kalimat: ";
        cin.ignore(); // Mengabaikan newline character
        getline(cin, input);

        if (isPalindrome(input)) {
            cout << input << " adalah kalimat palindrom.\n";
        } else {
            cout << input << " bukan kalimat palindrom.\n";
        }
    } else if (pilihan == 2) {
        break;
    } else {
        cout << "Pilihan tidak valid. Silakan masukkan 1 atau 2.\n";
    }
} while (true);

return 0;
}

```

Screenshot Output

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```

PS D:\C++\Data structure> & 'c:\Users\HP\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-syifhvu.d3b' '--stdout=Microsoft-MIEngine-Out-kkvzvbyq.uhe' '--stderr=Microsoft-MIEngine-Error-rx0dgvu5.a2c' '--pid=Microsoft-MIEngine-Pid-rsy14cbc.l3k' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'

Menu Program Pemeriksa Kalimat Palindrom:
1. Masukkan Kata/Kalimat
2. Exit
Pilihan Anda: 1
Masukkan kata/kalimat: Telkom
Telkom bukan kalimat palindrom.

Menu Program Pemeriksa Kalimat Palindrom:
1. Masukkan Kata/Kalimat
2. Exit
Pilihan Anda: 1
Masukkan kata/kalimat: Ini
Ini adalah kalimat palindrom.

Menu Program Pemeriksa Kalimat Palindrom:
1. Masukkan Kata/Kalimat
2. Exit
Pilihan Anda: 2
PS D:\C++\Data structure>
  
```

C/C++: ... ✓
 cppdbg: un...

IDENTITAS
 File Edit View
 NAMA : MUHAMAD NASRULLOH
 NIM : 2311102044
 KELAS : IF 11-B

Deskripsi

Program diatas sebuah implementasi sederhana dari stack di c++. Saya membuatnya dengan menu yang berisi 2 opsi : input kata/kalimat yang diinginkan oleh user serta exit (jika sudah selesai). Dengan 2 fungsi yaitu isPalindrome didalamnya untuk mengubah semua huruf menjadi huruf kecil (agar outputnya sama), membuat stack untuk menampung karakter kalimat, memasukkan karakter kalimat satu persatu dalam stack dan membandingkan karakter/inputan dari depan dan belakang (jika tidak sama maka kalimat bukan palindrom dan mengembalikan nilai false, jika sama kalimat adalah palindrom dan fungsi bernilai true); fungsi main () ini dengan adanya looping do while, user dapat menginputkan lagi untuk mengecek kata lainnya apakah palindrom atau tidak sampai dia memilih opsi exit.

- Unguided II

Buatlah program untuk melakukan pembalikan terhadap kalimat menggunakan stack dengan minimal 3 kata. Jelaskan output program dan source codenya beserta operasi/fungsi yang dibuat.

Source Code

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

using namespace std;

void reverseSentence(string str) {
    stack<char> s; // Deklarasi stack untuk menyimpan karakter kalimat

    for (char ch : str) { // Memasukkan karakter kalimat ke stack
        s.push(ch);
    }

    while (!s.empty()) { // Mengeluarkan karakter dari stack dan menampilkan
        cout << s.top();
        s.pop();
    }
}

int main() {
    string input;
    int pilihan;

    do {
```

```

cout << "\nMenu Program Pembalik Kalimat:\n";
cout << "1. Masukkan Kalimat\n";
cout << "2. Exit\n";
cout << "Pilihan Anda: ";
cin >> pilihan;

if (pilihan == 1) {
    cout << "Masukkan kalimat (minimal 3 kata): ";
    cin.ignore(); // Mengabaikan newline character
    getline(cin, input);

    // Hitung jumlah kata dalam kalimat
    int wordCount = 1;
    for (int i = 0; i < input.length(); i++) {
        if (input[i] == ' ') {
            wordCount++;
        }
    }

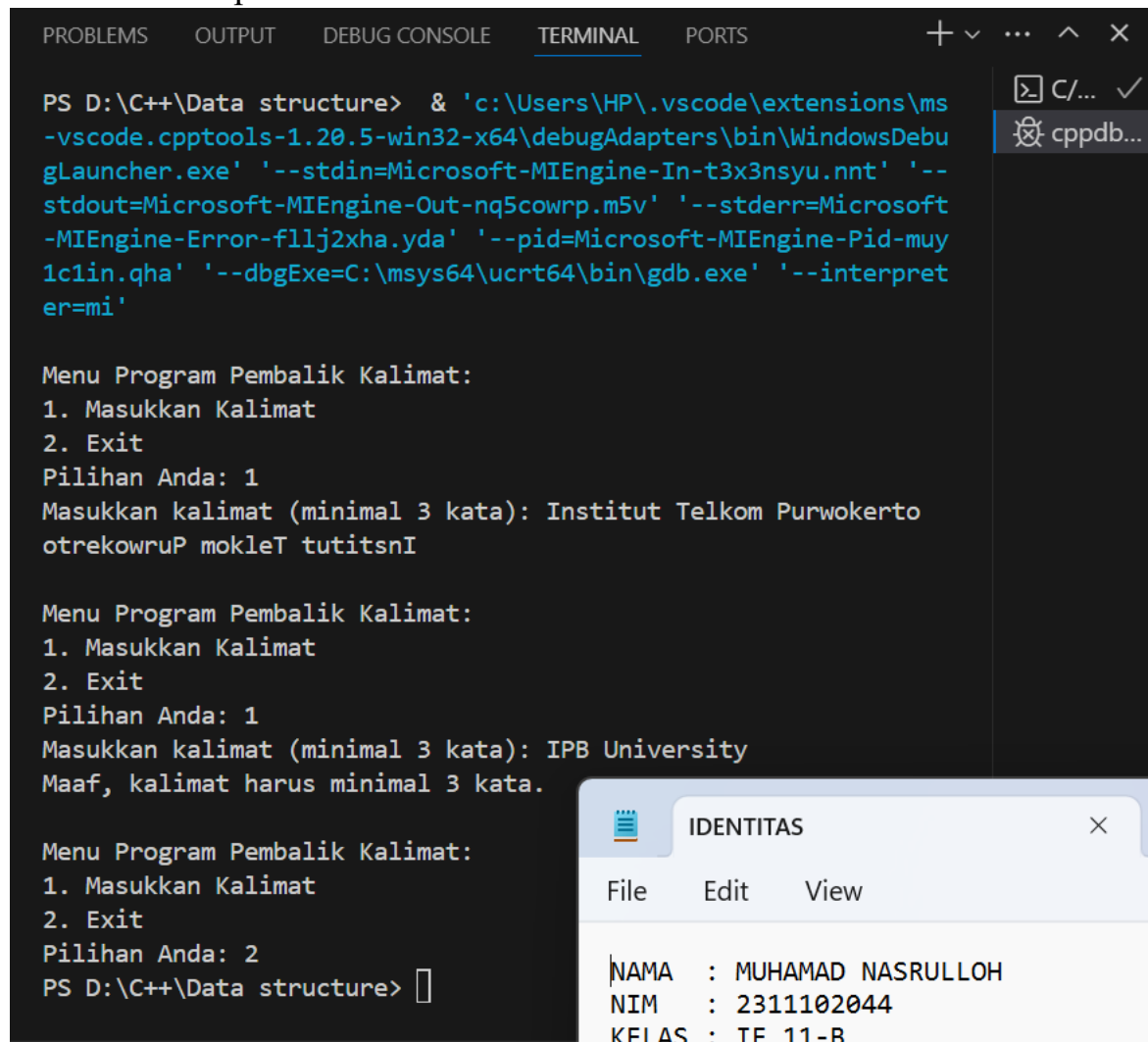
    // Periksa apakah kalimat memiliki minimal 3 kata
    if (wordCount < 3) {
        cout << "Maaf, kalimat harus minimal 3 kata.\n";
        continue;
    }

    reverseSentence(input);
    cout << endl;
} else if (pilihan == 2) {
    break;
} else {
    cout << "Pilihan tidak valid. Silakan masukkan 1 atau 2.\n";
}
} while (true);

return 0;
}

```

Screenshot Output



```
PS D:\C++\Data structure> & 'c:\Users\HP\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-t3x3nsyu.nnt' '--stdout=Microsoft-MIEngine-Out-nq5cowrp.m5v' '--stderr=Microsoft-MIEngine-Error-fllj2xha.yda' '--pid=Microsoft-MIEngine-Pid-muy1c1in.qha' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
```

Menu Program Pembalik Kalimat:
1. Masukkan Kalimat
2. Exit
Pilihan Anda: 1
Masukkan kalimat (minimal 3 kata): Institut Telkom Purwokerto otrekowruP mokleT tutitsnI

Menu Program Pembalik Kalimat:
1. Masukkan Kalimat
2. Exit
Pilihan Anda: 1
Masukkan kalimat (minimal 3 kata): IPB University
Maaf, kalimat harus minimal 3 kata.

Menu Program Pembalik Kalimat:
1. Masukkan Kalimat
2. Exit
Pilihan Anda: 2
PS D:\C++\Data structure> █

IDENTITAS

File Edit View

NAMA : MUHAMAD NASRULLOH
NIM : 2311102044
KELAS : IF 11-B

Deskripsi

Program diatas adalah implementasi stack di c++ yang sedikit lebih kompleks dibandingkan unguided 1 yang telah saya buat sebelumnya. Ada 2 fungsi yaitu reverseSentence didalamnya mendeklarasikan stack s untuk menyimpan karakter kalimat, lalu memasukkan karakter/inputan itu ke dalam stack s.push(ch) dengan looping for while (jika stack itu terisi maka akan mengeluarkan karakter/inputan tadi dan menampilkannya secara terbalik); di fungsi main () dengan opsi 1 user dimintai input kalimat lalu program akan membaca dan menghitung jumlah katanya dengan looping (jika dijumpai kurang dari 3 kata akan ditampilkan pesan error) dan jika sudah memenuhi 3 kata akan memanggil fungsi reverseSentence untuk membalik kalimat. Opsi 2 akan mengakhiri program karena user memilih exit/keluar

D. KESIMPULAN

Stack seperti yang dijelaskan pada dasar teori menerapkan konsep LIFO (Last In First Out) berarti elemen terakhir masuk adalah elemen yang pertama kali keluar. Dalam pemrograman tentu memiliki sejumlah kegunaan seperti untuk menyimpan status program sebelumnya sehingga dapat membatalkan operasi, mengevaluasi ekspresi matematika dan boolean, mencocokkan tanda kurung dan digunakan untuk menerapkan algoritma rekursif dan non-rekursif. Adapun secara kelebihanannya, stack mudah dipahami lebih efisien untuk operasi push dan popnya serta serbaguna (dapat digunakan untuk bermacam aplikasi). Kekurangannya terbatas (hanya dapat mengakses data/inputan paling atas) dan terkadang menggunakan banyak memori.

E. REFERENSI

- [1] Kabir Singh (2024). Stack in C++
<https://www.naukri.com/code360/library/stack-in-c-stl>.
- [2] Trivusi (2022). Struktur Data Stack : Pengertian, Karakteristik dan Kegunaannya
<https://www.trivusi.web.id/2022/07/struktur-data-stack.html>.
- [3] Asisten Praktikum (2024). Modul 6 Stack