

LAPORAN PRAKTIKUM

Modul VI STACK(TUMPUKAN)



Disusun oleh:

Muhammad Irsyad : **2211102048**

Dosen

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

**PROGRAM STUDI S1 INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2023**

BAB I

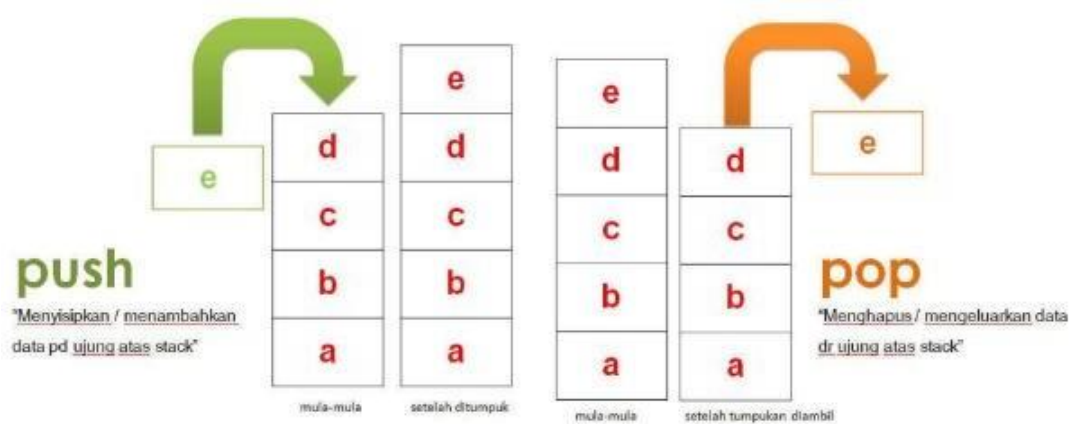
Tujuan Pembelajaran

1. Memahami konsep dan implementasi stack: Tujuan utama adalah untuk memahami konsep dasar dari struktur data stack dan bagaimana cara mengimplementasikannya dalam bahasa pemrograman C++.
2. Menggunakan stack dalam pemecahan masalah: Stack adalah struktur data yang berguna dalam berbagai pemecahan masalah dan algoritma.
3. Memahami konsep LIFO (Last-In, First-Out): Stack mengikuti prinsip LIFO, yang berarti elemen terakhir yang dimasukkan ke dalam stack akan menjadi elemen pertama yang dikeluarkan.
4. Memahami kompleksitas waktu operasi stack: Dalam pembelajaran stack, Anda akan mempelajari kompleksitas waktu operasi dasar seperti push, pop, dan top.
5. Mengembangkan keterampilan pemrograman C++: Pembelajaran stack dalam bahasa C++ akan membantu Anda memperkuat pemahaman Anda tentang bahasa pemrograman C++ itu sendiri.

BAB II

Dasar Teori

Stack atau tumpukan merupakan sebuah kumpulan data yang diletakan di atas data lain, seperti tumpukan rak buku. Satu hal yang perlu kita ingat adalah bahwa kita bisa menambah data dan mengambil data lewat ujung yang sama, yang disebut sebagai ujung atas tumpukan (top of stack). Lifo(last in First Out) adalah sifat dari Stack data yang di simpan terakhir akan di ambil lebih terdahulu, data yang disimpan pertama kali akan diambil paling akhir



Operasi Stack, Push and Pop

Operasi Stack ada banyak yaitu

- Push
Mengisi data ke atas stack (tumpukan).
- Pop
Menghapus/Menghilangkan satu data ditumpukan teratas.
- isFull
Memeriksa apakah ruang stack sudah penuh atau belum.
- isEmpty
Memeriksa apakah ruang stack kosong atau tidak.
- Peek
Mencek data diposisi tertentu.
- Count
Menghitung banyak data pada stack.
- Change
Mengubah posisi data ke posisi tertentu.
- Display
Menampilkan semua data pada stack
- Destroy
Menghapus atau membersihkan semua data pada stack

BAB III

LATIHAN KELAS – GUIDED

Guided 1

Source Code

```
#include <iostream>
using namespace std;
string arrayBuku[5];
int maksimal = 5, top = 0;
bool isFull()
{
    if (top == maksimal)
    {
        return true;
    }
    else
    {
        return false;
    }
}
bool isEmpty()
{
    if (top == 0)
    {
        return true;
    }
    else
    {
        return false;
    }
}
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; i >= 0; i--)
        {
            cout << arrayBuku[i] << endl;
        }
    }
}

int main()
{
    pushArrayBuku("Kalkulus");
    pushArrayBuku("Struktur Data");
    pushArrayBuku("Matematika Distit");
    pushArrayBuku("Dasar Multimedia");
    pushArrayBuku("Inggris");
    cetakArrayBuku();
    cout << "\n" << endl;
    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" << endl;
    destroyArraybuku();
    cout << top;
    cetakArrayBuku();
}

```

Screenshot Program

```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Insyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2> & 'c:\Users\Insyad\.vscode\extensions\ms-vscode.cpptools-1.15.4-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-wudtkgob.tdn' '--stdout=Microsoft-MIEngine-Out-mwqhowxq.wev' '--stderr=Microsoft-MIEngine-Error-1jlg5vuq.jfq' '--pid=Microsoft-MIEngine-Pid-fvnhdaut.huj' '--dbgExe=c:\MinGW\bin\gdb.exe' '--interpreter=mi'

Inggris
Dasar Multimedia
Matematika Distit
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

0tidak ada data yang dicetak
PS C:\Users\Insyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2> 
```

Deskripsi Program

Program di atas merupakan implementasi dari struktur data stack menggunakan array. Berikut adalah deskripsi singkat dari setiap fungsi dan bagian programnya:

- Fungsi ``isFull()``: Fungsi ini digunakan untuk memeriksa apakah stack sudah penuh atau belum. Jika ``top`` (indeks teratas stack) sama dengan ``maksimal`` (ukuran maksimum stack), maka stack dianggap penuh dan fungsi ini mengembalikan nilai ``true``, jika tidak, maka fungsi ini mengembalikan nilai ``false``.
- Fungsi ``isEmpty()``: Fungsi ini digunakan untuk memeriksa apakah stack kosong atau tidak. Jika ``top`` sama dengan 0, maka stack dianggap kosong dan fungsi ini mengembalikan nilai ``true``, jika tidak, maka fungsi ini mengembalikan nilai ``false``.

- Fungsi `pushArrayBuku(string data)`: Fungsi ini digunakan untuk menambahkan data (`data`) ke stack. Jika stack penuh (memanggil fungsi `isFull()` mengembalikan `true`), maka akan ditampilkan pesan "data telah penuh". Jika stack tidak penuh, maka data akan ditambahkan ke elemen teratas stack (`arrayBuku[top]`) dan `top` akan dinaikkan.

- Fungsi `popArrayBuku()`: Fungsi ini digunakan untuk menghapus data dari stack. Jika stack kosong (memanggil fungsi `isEmpty()` mengembalikan `true`), maka akan ditampilkan pesan "tidak ada data yang dihapus". Jika stack tidak kosong, maka elemen teratas stack akan dihapus dengan mengosongkan nilai (`arrayBuku[top - 1]`) dan `top` akan dikurangi.

- Fungsi `peekArrayBuku(int posisi)`: Fungsi ini digunakan untuk melihat data pada posisi tertentu dalam stack. Jika stack kosong, akan ditampilkan pesan "tidak ada data yang bisa dilihat". Jika stack tidak kosong, maka akan ditemukan indeks elemen pada posisi yang diinginkan dengan mengurangi `top` dengan `posisi`. Kemudian, data pada posisi tersebut (`arrayBuku[index]`) akan ditampilkan.

- Fungsi `countStack()`: Fungsi ini mengembalikan jumlah elemen dalam stack, yaitu nilai `top`.

- Fungsi `changeArrayBuku(int posisi, string data)`: Fungsi ini digunakan untuk mengubah data pada posisi tertentu dalam stack dengan data baru (`data`). Jika `posisi` melebihi nilai `top`, maka akan ditampilkan pesan "posisi melebihi data yang ada". Jika tidak, data pada posisi tersebut (`arrayBuku[index]`) akan diubah menjadi `data`.

- Fungsi `destroyArrayBuku()`: Fungsi ini digunakan untuk mengosongkan seluruh data dalam stack dengan mengosongkan semua elemen array `arrayBuku` dan mengatur `top` menjadi 0.

- Fungsi ``cetakArrayBuku()``: Fungsi ini digunakan untuk mencetak semua data dalam stack dari elemen teratas ke elemen terbawah. Jika stack kosong, akan ditampilkan pesan "tidak ada data yang dicetak". Jika tidak, semua elemen dalam ``arrayBuku`` akan dicetak dari elemen teratas (```

`arrayBuku[i]``) hingga elemen terbawah.

- Di dalam fungsi ``main()``, terdapat penggunaan fungsi-fungsi di atas untuk melakukan operasi pada stack. Beberapa operasi yang dilakukan antara lain: menambahkan data ke stack menggunakan ``pushArrayBuku()``, mencetak data dalam stack menggunakan ``cetakArrayBuku()``, memeriksa apakah stack penuh menggunakan ``isFull()``, memeriksa apakah stack kosong menggunakan ``isEmpty()``, melihat data pada posisi tertentu menggunakan ``peekArrayBuku()``, menghapus data dari stack menggunakan ``popArrayBuku()``, mengubah data pada posisi tertentu dalam stack menggunakan ``changeArrayBuku()``, menghitung jumlah elemen dalam stack menggunakan ``countStack()``, mengosongkan stack menggunakan ``destroyArrayBuku()``, dan mencetak data dalam stack setelah dihapus menggunakan ``cetakArrayBuku()``.

TUGAS – UNGUIDED

Unguided 1

Source Code

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

using namespace std;

string reverseString(string input)
{
    stack<char> charStack;
    string reversedString;
    for (char c : input)
    {
        charStack.push(c);
    }
    while (!charStack.empty())
    {
        reversedString += charStack.top();
        charStack.pop();
    }
    return reversedString;
}

int main()
{
    string input;
    for (int i = 0; i < 5; i++)
    {
        cout << "Masukkan string: ";
        getline(cin, input);

        string reversed = reverseString(input);
```

```

        cout << "String terbalik: " << reversed << endl;
    }
    return 0;
}

```

Screenshot Program

```

PS C:\Users\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2> & 'c:\Users\Irsyad\.vscode\extensions\ms-vscode.cpptools-1.15.4-win32-x64\debugAdapters\bin\windowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-evj5mgdb.pko' '--stdout=Microsoft-MIEngine-Out-bg2i4iah.04a' '--stderr=Microsoft-MIEngine-Error-4dwhnso.2wa' '--pid=Microsoft-MIEngine-Pid-5bgjnnuc.s44' '--dbgExe=C:\MinGW\bin\gdb.exe' '--interpreter=mi'
Masukkan string: Telkom Purwokerto
String terbalik: otrekowruP mokleT
Masukkan string: Muhammad Irsyad
String terbalik: dayrI dammahuM
Masukkan string: Teknik Informatika
String terbalik: akitamrofni kinkeT
Masukkan string: Alstrukdat
String terbalik: tadkurtsIA
Masukkan string: Kasur
String terbalik: rusaK
PS C:\Users\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2>

```

Deskripsi Program

Program di atas merupakan implementasi untuk membalikkan string menggunakan stack. Berikut adalah deskripsi singkat dari setiap fungsi dan bagian programnya:

- Fungsi ``reverseString(string input)``: Fungsi ini digunakan untuk membalikkan string yang diberikan sebagai input. Fungsi ini menggunakan stack untuk membalikkan string. Pertama, fungsi ini membuat sebuah stack bernama ``charStack`` yang berisi karakter. Kemudian, setiap karakter dalam string ``input`` dimasukkan ke dalam stack menggunakan perulangan. Setelah itu, fungsi mengambil karakter teratas dari stack menggunakan ``charStack.top()``, menambahkannya ke string ``reversedString``, dan menghapus karakter tersebut dari stack menggunakan ``charStack.pop()``. Hal ini diulang hingga stack kosong. Akhirnya, fungsi mengembalikan string ``reversedString`` yang merupakan hasil string yang telah dibalik.

- Fungsi ``main()``: Fungsi ini merupakan fungsi utama program. Di dalamnya, pengguna diminta untuk memasukkan sebuah string menggunakan ``getline(cin, input)``. Kemudian, fungsi ``reverseString()`` dipanggil dengan mengirimkan string

`input` sebagai argumen. Hasil yang dikembalikan dari fungsi `reverseString()` disimpan dalam variabel `reversed`. Akhirnya, string terbalik (`reversed`) dicetak ke layar.

Program ini menggunakan library ``<stack>`` untuk mengimplementasikan stack dan ``<string>`` untuk menggunakan tipe data string. String yang dimasukkan oleh pengguna akan diproses dan dibalik menggunakan stack, kemudian hasilnya dicetak ke layar.

Unguided 2

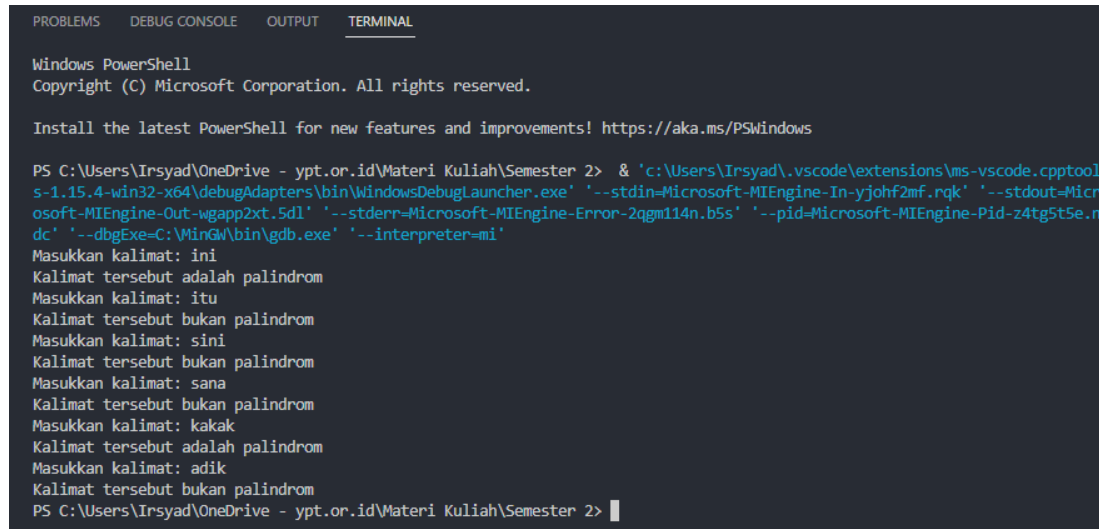
Source Code

```
#include <iostream>
#include <stack>
#include <string>
using namespace std;
bool isPalindrome(string sentence)
{
    stack<char> charStack;
    int length = sentence.length();

    for (int i = 0; i < length; i++)
    {
        charStack.push(sentence[i]);
    }
    for (int i = 0; i < length; i++)
    {
        if (sentence[i] != charStack.top())
        {
            return false;
        }
        charStack.pop();
    }
    return true;
}
```

```
int main()
{
    string sentence;
    for (int i = 0; i < 5; i++)
    {
        cout << "Masukkan kalimat: ";
        getline(cin, sentence);
        if (isPalindrome(sentence))
        {
            cout << "Kalimat tersebut adalah palindrom" << endl;
        }
        else
        {
            cout << "Kalimat tersebut bukan palindrom" << endl;
        }
    }
    cout << "Masukkan kalimat: ";
    getline(cin, sentence);
    if (isPalindrome(sentence))
    {
        cout << "Kalimat tersebut adalah palindrom" << endl;
    }
    else
    {
        cout << "Kalimat tersebut bukan palindrom" << endl;
    }
    return 0;
}
```

Screenshot Program



```
PROBLEMS  DEBUG CONSOLE  OUTPUT  TERMINAL

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Install the latest PowerShell for new features and improvements! https://aka.ms/PSWindows

PS C:\Users\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2> & 'c:\Users\Irsyad\.vscode\extensions\ms-vscode.cpptools-1.15.4-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-yjohf2mf.rqk' '--stdout=Microsoft-MIEngine-Out-wgapp2xt.5d1' '--stderr=Microsoft-MIEngine-Error-2qgm114n.b5s' '--pid=Microsoft-MIEngine-Pid-z4tg5tSe.n
dc' '--dbgExe=C:\MinGW\bin\gdb.exe' '--interpreter=mi'
Masukkan kalimat: ini
Kalimat tersebut adalah palindrom
Masukkan kalimat: itu
Kalimat tersebut bukan palindrom
Masukkan kalimat: sini
Kalimat tersebut bukan palindrom
Masukkan kalimat: sana
Kalimat tersebut bukan palindrom
Masukkan kalimat: kakak
Kalimat tersebut adalah palindrom
Masukkan kalimat: adik
Kalimat tersebut bukan palindrom
PS C:\Users\Irsyad\OneDrive - ypt.or.id\Materi Kuliah\Semester 2> |
```

Deskripsi Program

Program di atas merupakan implementasi untuk membalikkan string menggunakan stack. Berikut adalah deskripsi singkat dari setiap fungsi dan bagian programnya:

- Fungsi `reverseString(string input)`: Fungsi ini digunakan untuk membalikkan string yang diberikan sebagai input. Fungsi ini menggunakan stack untuk membalikkan string. Pertama, fungsi ini membuat sebuah stack bernama `charStack` yang berisi karakter. Kemudian, setiap karakter dalam string `input` dimasukkan ke dalam stack menggunakan perulangan. Setelah itu, fungsi mengambil karakter teratas dari stack menggunakan `charStack.top()`, menambahkannya ke string `reversedString`, dan menghapus karakter tersebut dari stack menggunakan `charStack.pop()`. Hal ini diulang hingga stack kosong. Akhirnya, fungsi mengembalikan string `reversedString` yang merupakan hasil string yang telah dibalik.

- Fungsi `main()`: Fungsi ini merupakan fungsi utama program. Di dalamnya, pengguna diminta untuk memasukkan sebuah string menggunakan `getline(cin, input)`. Kemudian, fungsi `reverseString()` dipanggil dengan mengirimkan string `input` sebagai argumen. Hasil yang dikembalikan dari fungsi `reverseString()`

disimpan dalam variabel ``reversed``. Akhirnya, string terbalik (``reversed``) dicetak ke layar.

Program ini menggunakan library ``<stack>`` untuk mengimplementasikan stack dan ``<string>`` untuk menggunakan tipe data string. String yang dimasukkan oleh pengguna akan diproses dan dibalik menggunakan stack, kemudian hasilnya dicetak ke layar.

BAB IV

KESIMPULAN

Dalam kedua di atas, dapat disimpulkan bahwa penggunaan stack sangat berguna dalam beberapa situasi pemrograman.

Pada contoh pertama, program menggunakan array sebagai struktur data dasar untuk mengimplementasikan stack. Berbagai operasi stack seperti push (penambahan elemen), pop (penghapusan elemen), peek (melihat elemen teratas), dan lainnya diimplementasikan dengan memanipulasi array. Program ini memberikan fleksibilitas dalam mengatur ukuran stack sesuai kebutuhan, namun juga memerlukan perhatian untuk memastikan batasan ukuran stack terjaga dan operasi dilakukan dengan benar.

Pada contoh kedua, program menggunakan struktur data stack yang disediakan oleh library ``<stack>`` dalam bahasa C++. Program ini memanfaatkan stack untuk membalikkan string. Dalam hal ini, penggunaan stack menyederhanakan proses pembalikan string, karena karakter-karakter string dapat dengan mudah dimasukkan ke dalam stack dan diambil kembali dalam urutan terbalik. Program ini juga memberikan kelebihan seperti pengelolaan otomatis dari struktur data stack dan ketahanan terhadap kesalahan operasi stack.

Kesimpulannya, stack merupakan struktur data yang penting dan serbaguna dalam pemrograman. Dalam implementasinya, baik menggunakan array maupun struktur data stack yang disediakan oleh library, stack dapat digunakan untuk menyelesaikan berbagai masalah dan tugas pemrograman seperti pengelolaan tumpukan data, pembalikan urutan data, evaluasi ekspresi matematika, dan lainnya. Penggunaan stack secara tepat dapat meningkatkan efisiensi dan kejelasan kode program serta memudahkan dalam pemecahan masalah.