

**LAPORAN PRAKTIKUM**  
**PERTEMUAN 7**  
**MODUL 7 STD**



**Nama :**

Farrel I'zaz Yuwono (2311104014)

**Dosen :**

YUDHA ISLAMI SULISTYA

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**  
**FAKULTAS INFORMATIKA**  
**TELKOM UNIVERSITY PURWOKERTO**  
**2024**

## TUGAS PENDAHULUAN

### Stack.h

```
#ifndef STACK_H
#define STACK_H

typedef char infotype;

struct stack {
    infotype info[15];
    int Top;
};

void createStack(stack &S);
bool isEmpty(stack S);
bool isFull(stack S);
void push(stack &S, infotype x);
infotype pop(stack &S);
void printInfo(stack S);

#endif
```

### Stack.cpp

```
#include "stack.h"
#include <iostream>
using namespace std;

void createStack(stack &S) {
    S.Top = 0;
}
```

```
bool isEmpty(stack S) {  
    return S.Top == 0;  
}  
  
bool isFull(stack S) {  
    return S.Top == 15;  
}  
  
void push(stack &S, infotype x) {  
    if (!isFull(S)) {  
        S.Top++;  
        S.info[S.Top] = x;  
    } else {  
        cout << "Stack penuh!" << endl;  
    }  
}  
  
infotype pop(stack &S) {  
    if (!isEmpty(S)) {  
        infotype x = S.info[S.Top];  
        S.Top--;  
        return x;  
    } else {  
        cout << "Stack kosong!" << endl;  
        return '\0';  
    }  
}
```

```

void printInfo(stack S) {
    if (!isEmpty(S)) {
        for (int i = S.Top; i >= 1; i--) {
            cout << S.info[i] << " ";
        }
        cout << endl;
    } else {
        cout << "Stack kosong!" << endl;
    }
}

```

Main.cpp

```

#include "stack.h"
#include <iostream>
using namespace std;

void fillStack(stack &S, const string &text) {
    for (char ch : text) {
        if (ch != ' ') push(S, ch);
    }
}

void displayStack(stack &S, int mod) {
    cout << "Isi stack awal: ";
    printInfo(S);
}

```

```
if (mod == 0) {  
    cout << "I F L A B J A Y A" << endl;  
} else if (mod == 1) {  
    cout << "H A L O B A N D U N G" << endl;  
} else if (mod == 2) {  
    cout << "P E R C A Y A D I R I" << endl;  
} else if (mod == 3) {  
    cout << "S T R U K T U R D A T A" << endl;  
}
```

```
cout << "Isi stack setelah pop: ";  
for (int i = 0; i < 4; ++i) pop(S);  
printInfo(S);  
}
```

```
int main() {  
    stack S;  
    createStack(S);  
    int mod = 0414;  
  
    if (mod == 0) fillStack(S, "IFLABJAYA");  
    else if (mod == 1) fillStack(S, "HALOBANDUNG");  
    else if (mod == 2) fillStack(S, "PERCAYADIRI");  
    else if (mod == 3) fillStack(S, "STRUKTURDATA");  
  
    displayStack(S, mod);  
    return 0;  
}
```

Output :

```
PS D:\kuliah\strukturdat\tugas kuliah\07_Stack\TP> ./TP
Masukkan digit terakhir NIM Anda: 4
Isi stack awal: A Y A J B A L F I
Output:
I F L A B J A Y A
J A Y A
Isi stack setelah pop: B A L F I
PS D:\kuliah\strukturdat\tugas kuliah\07_Stack\TP> █
```

Penjelasan kode program :

1. File stack.h:

Berfungsi sebagai header yang mendefinisikan struktur stack dan deklarasi fungsi-fungsi untuk operasi dasar stack.

Struktur stack memiliki array info untuk menyimpan elemen stack (maksimal 15 elemen) dan variabel Top untuk menunjukkan posisi elemen teratas.

2. File stack.cpp:

Berisi implementasi fungsi-fungsi operasi stack, seperti:

- a. createStack: Menginisialisasi stack sebagai kosong (Top = 0).
- b. isEmpty: Mengecek apakah stack kosong.
- c. isFull: Mengecek apakah stack penuh.
- d. push: Menambahkan elemen ke stack jika belum penuh.
- e. pop: Menghapus elemen teratas dari stack jika tidak kosong.
- f. printInfo: Menampilkan seluruh elemen dalam stack dari atas ke bawah.

3. File main.cpp:

- a. Program utama yang meminta pengguna memasukkan digit terakhir NIM mereka.
- b. Menghitung mod dari digit terakhir NIM dengan 4 untuk menentukan kata yang akan dimasukkan ke stack.

- c. Mengisi stack dengan karakter-karakter dari kata yang dipilih.
- d. Menampilkan isi stack sebelum dan sesudah melakukan empat kali operasi pop untuk memperlihatkan perubahan isi stack.

## UNGUIDED

### 1. Soal Palindrom

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

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

    for (char c : str) {
        if (isalpha(c)) {
            s.push(tolower(c));
        }
    }

    for (char c : str) {
        if (isalpha(c)) {
            if (tolower(c) != s.top()) {
                return false;
            }
            s.pop();
        }
    }
}
```

```

    }

    return true;
}

int main() {
    string kalimat;

    cout << "Masukkan kalimat: ";

    getline(cin, kalimat);

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

    return 0;
}

```

Output :

```

PS D:\kuliah\strukturdat\tugas kuliah\07_Stack> cd "d:
d_01.cpp -o unguided_01 } ; if ($?) { .\unguided_01 }
Masukkan kalimat: malam
Kalimat adalah palindrom.
PS D:\kuliah\strukturdat\tugas kuliah\07_Stack>

```



## 2. Pembalik Kata

```
#include <iostream>

#include <stack>

#include <sstream>

using namespace std;

void reverseEachWord(const string &str) {

    stringstream ss(str);

    string word;

    cout << "Data stack Array:\nData : ";

    while (ss >> word) {

        stack<char> s;

        for (char ch : word) {

            s.push(ch);

        }

        while (!s.empty()) {

            cout << s.top();

            s.pop();

        }

        cout << " ";

    }

    cout << endl;

}
```

```

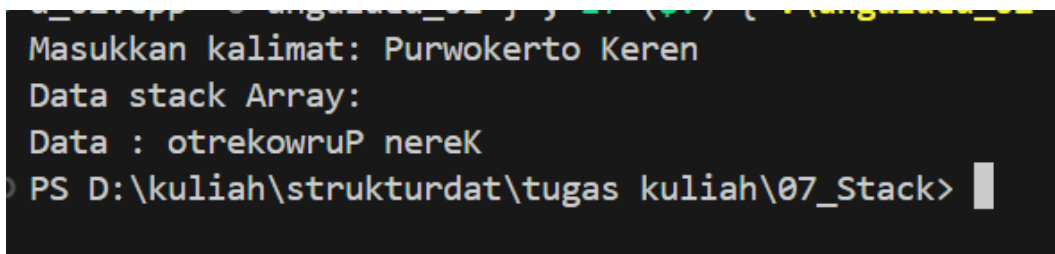
int main() {
    string kalimat;
    cout << "Masukkan kalimat: ";
    getline(cin, kalimat);

    reverseEachWord(kalimat);

    return 0;
}

```

Output :



```

Masukkan kalimat: Purwokerto Keren
Data stack Array:
Data : otrekowruP nereK
PS D:\kuliah\strukturdat\tugas kuliah\07_Stack>

```

Penjelasan Kode Program :

1. Stack digunakan untuk membalik urutan karakter dari kalimat asli.

Setelah memasukkan karakter ke stack, kita membandingkan karakter dari depan dan belakang.

Jika semua karakter sesuai saat dibaca dari depan dan belakang, maka kalimat tersebut adalah palindrom.

2. Menggunakan stringstream untuk Membagi Kata: Program menggunakan stringstream untuk membagi input menjadi kata-kata.

Stack untuk Membalik Huruf dalam Kata: Setiap huruf dari kata dimasukkan ke dalam stack. Karena stack menggunakan prinsip LIFO (Last In, First Out), ketika kita mengeluarkan huruf-hurufnya, urutannya akan menjadi terbalik.

Menampilkan Kata yang Sudah Dibalik: Setelah huruf dalam satu kata dibalik, program menampilkan kata tersebut diikuti dengan spasi.