

**LAPORAN PRAKTIKUM STRUKTUR  
DATA DAN ALGORITMA**

**MODUL VI  
STACK**



**Disusun Oleh :**

NAMA : D'sharlendita Febianda Aurelia  
NIM : 2311102069

**Dosen :**

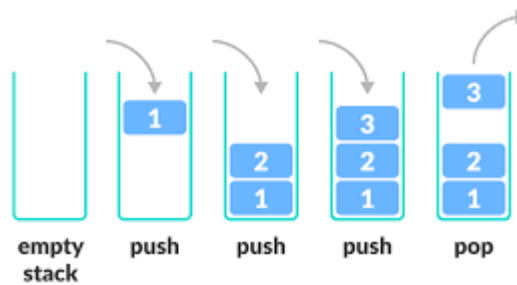
Wahyu Andi Saputra, S.pd., M,Eng

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

## A. Dasar Teori

### 1. Pengertian Stack

Stack adalah data linier yang mengikuti prinsip Last – In – First – Out (LIFO).



**Gambar A.1** *Stack*

### 2. Operasi Kunci pada Struktur Data Tumpukan

- Push : Menambahkan elemen ke bagian atas Stack
- Pop : Menghapus elemen teratas dari Stack
- Peek : Mengembalikan elemen teratas tanpa menghapusnya
- IsEmpty : Memeriksa apakah Stack kosong
- IsFull : Memeriksa apakah Stack sudah penuh
- Size: Mengembalikan jumlah elemen yang ada dalam Stack.
- Peek :Melihat nilai atau elemen pada posisi tertentu dalam Stack tanpa menghapusnya.
- Clear : Mengosongkan atau menghapus semua elemen dari Stack.
- Search : Mencari keberadaan elemen tertentu dalam Stack.

### 3. Kelebihan dan Kekurangan Menggunakan Stack

#### a. Kelebihan

- Membantu mengelola data dengan metode LIFO
- Secara otomatis membersihkan objek
- Tidak mudah rusak

- Ukuran variabel tidak dapat diubah
- Mengontrol memori secara mandiri

b. Kekurangan

- Memori stack sangat terbatas
- Ada kemungkinan stack akan meluap atau overflow jika terlalu banyak objek
- Tidak memungkinkan akses acak, karena harus mengeluarkan tumpukan paling atas terlebih dahulu untuk mengakses tumpukan paling bawah

## B. Guided

### Guided 1

Program Stack

Source Code:

```
//D'sharlendita Febianda Aurelia
//2311102069

#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;
}

```

Screenshots Output :

The screenshot shows the output of a C++ program in a terminal window. 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). It changes the element at index 2 to Bahasa Jerman and prints the stack again. Finally, it destroys the stack and prints the top element (0) and the number of elements (0). The terminal prompt is PS C:\parktikum struktur data\MODUL 6\guided>.

Overlaid on the terminal output is a Notepad++ window titled 'Na'. The window contains the following text:

```

Nama : D'sharlendita Febianda Aurelia
NIM : 2311102069
Kelas : IF - 11 - B

```

The Notepad++ window also shows the status bar at the bottom: Ln 1, Col 7 | 77 characters | 100% | Window | UTF-8.

Deskripsi:

Pengguna dapat menambahkan, menghapus, melihat, mengubah, dan

menghitung buku dalam tumpukan. Fungsi-fungsi seperti pushArrayBuku(), popArrayBuku(), peekArrayBuku(), changeArrayBuku(), countStack(), destroyArrayBuku(), dan cetakArrayBuku() disediakan untuk mengelola tumpukan buku.

## C. Unguided

### 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.

Source Code :

```
//D'sharlendita Febianda Aurelia
//2311102069

#include <iostream>
#include <stack>
#include <string>
#include <locale>

using namespace std;

// Fungsi untuk menghapus karakter non-alphanumeric dari sebuah string
string removeNonAlphanumeric(string str) {
    string result = "";
    for (char c : str) {
        if (isalnum(c)) { // isalnum() memeriksa apakah karakter
            // adalah alfanumerik
            result += tolower(c); // Mengubah karakter menjadi
            // huruf kecil
        }
    }
    return result;
}

// Fungsi untuk menentukan apakah sebuah string adalah palindrom
bool isPalindrome(string str) {
    stack<char> charStack;
```

```

    int length = str.length();

    // Menambahkan setengah karakter pertama ke stack
    for (int i = 0; i < length / 2; i++) {
        charStack.push(str[i]);
    }

    // Memeriksa setengah karakter kedua dengan stack
    for (int i = (length + 1) / 2; i < length; i++) {
        if (charStack.top() != str[i]) {
            return false;
        }
        charStack.pop();
    }

    return true;
}

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

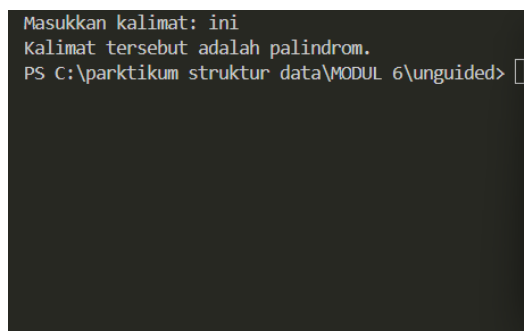
    // Menghapus karakter non-alphanumeric dan mengubah menjadi
    huruf kecil
    string cleanedInput = removeNonAlphanumeric(input);

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

    return 0;
}

```

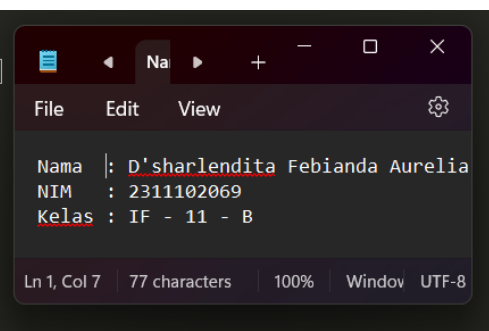
## Screenshots Output



```

Masukkan kalimat: ini
Kalimat tersebut adalah palindrom.
PS C:\parktikum struktur data\MODUL 6\unguided>

```



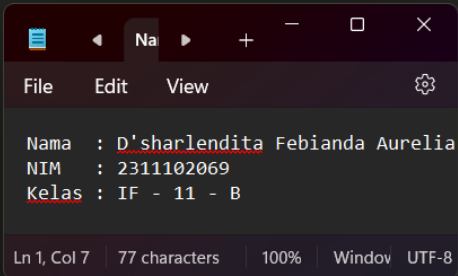
```

Nama   |: D'sharlendita Febianda Aurelia
NIM    |: 2311102069
Kelas |: IF - 11 - B

```



```
Masukkan kalimat: dita
Kalimat tersebut bukan palindrom.
PS C:\parktikum struktur data\MODUL 6\unguided>
```



The screenshot shows a Notepad++ window with a dark theme. The title bar says 'Na'. The menu bar has 'File', 'Edit', and 'View'. The text content is: 'Nama : D'sharlendita Febianda Aurelia', 'NIM : 2311102069', and 'Kelas : IF - 11 - B'. The status bar at the bottom shows 'Ln 1, Col 7', '77 characters', '100%', 'Window', and 'UTF-8'.

### Deskripsi:

Program ini menggunakan stack untuk mengecek apakah sebuah string adalah palindrom. Stack adalah struktur data yang hanya memungkinkan akses ke elemen teratas. Dalam program ini, stack digunakan untuk menyimpan setengah karakter pertama dari string. Kemudian, setengah karakter kedua dibandingkan dengan karakter yang disimpan di stack. Jika semua karakter cocok, string adalah palindrom.

### Unguided 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

Source Code:

```
//D'sharlendita Febianda Aurelia
//2311102069

#include <iostream>
#include <cstring>

using namespace std;

struct Tumpukan {
    char nama[50];
    int top;
} T;

void push(char x) {
```

```

        if (T.top == 49) { // Memeriksa apakah tumpukan sudah penuh
            cout << "Tumpukan sudah penuh";
        } else {
            T.top = T.top + 1;
            T.nama[T.top] = x; // Memasukkan karakter ke dalam
tumpukan
        }
    }

char pop() {
    char hasil;
    if (T.top == 0) { // Memeriksa apakah tumpukan sudah kosong
        cout << "Tumpukan sudah kosong";
        hasil = ' ';
    } else {
        hasil = T.nama[T.top]; // Mengambil karakter dari
tumpukan
        T.top = T.top - 1; // Mengurangi indeks top untuk
menunjuk ke elemen berikutnya
    }
    return hasil; // Mengembalikan karakter yang diambil dari
tumpukan
}

int main() {
    int len;
    char kalimat[25];
    T.top = 0;

    cout << "Masukkan Kalimat: ";
    cin.getline(kalimat, 25); // Menerima masukan kalimat dari
pengguna

    len = strlen(kalimat); // Menghitung panjang kalimat

    cout << "Kalimat Asli: " << kalimat; // Menampilkan kalimat
asli yang dimasukkan pengguna

    for (int i = 0; i < len; i++) {
        push(kalimat[i]); // Memasukkan setiap karakter kalimat
ke dalam tumpukan
    }
}

```

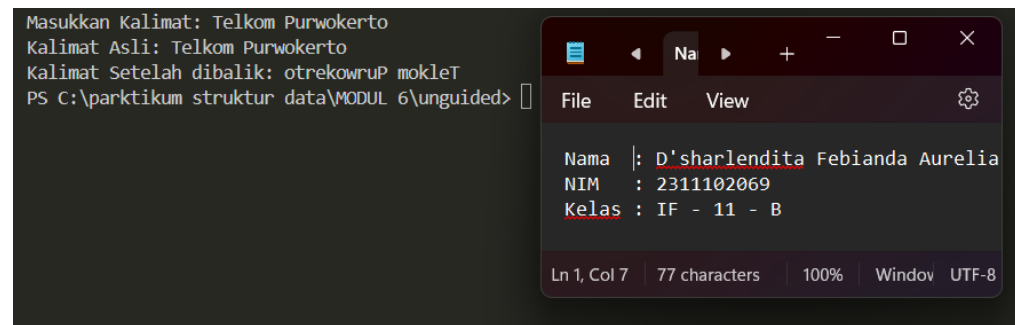
```

    cout << "\nKalimat Setelah dibalik: ";
    for (int i = 0; i < len; i++) {
        cout << pop(); // Mengambil dan menampilkan karakter dari
tumpukan untuk membalikkan kalimat
    }
    cout << endl;

    return 0;
}

```

Screenshot Output:



Deskripsi:

Program ini menggunakan struktur data tumpukan untuk membalikkan kalimat. Kalimat dimasukkan ke dalam tumpukan karakter demi karakter. Kemudian, karakter diambil dari tumpukan satu per satu dan ditampilkan, sehingga menghasilkan kalimat yang terbalik.

#### D. Kesimpulan

Praktikum Modul Stack memberikan pemahaman yang lebih mendalam mengenai struktur data stack dan penggunaannya dalam pemrograman. Melalui latihan-latihan yang diberikan, saya dapat memahami konsep stack, seperti LIFO (Last In, First Out), operasi dasar stack (push, pop, peek), dan berbagai aplikasi stack dalam menyelesaikan permasalahan pemrograman.

Saya memahami bahwa stack adalah struktur data linear yang menyimpan

elemen dengan urutan LIFO. Elemen yang terakhir dimasukkan adalah yang pertama dikeluarkan. dari operasi dasar stack saya dapat menerapkan operasi dasar stack, seperti push (memasukkan elemen), pop (menghapus elemen), peek (melihat elemen teratas), dan isEmpty (memeriksa apakah stack kosong). Dan juga saya dapat menggunakan stack untuk menyelesaikan berbagai permasalahan pemrograman, seperti pembalikan kalimat dan mengecek palindrom.

## **E. Referensi**

Asisten Praktikum. 15 Mei 2024. "Modul 6 Stack". Diakses pada 19 Mei 2024, dari Learning Management System. 2024

Kumar, Harendra. (2024, 02 Mei). Stack Data Structure. Diakses pada 20 Mei 2024, dari <https://www.geeksforgeeks.org/stack-data-structure/>

Adieb, Maulana. (2021, 04 Febuari). Memahami Stack, Struktur Data yang Mudahkan Programmer Jalankan Tugasnya. Diakses pada 20 Mei 2024, dari <https://glints.com/id/lowongan/stack-adalah/>