

# **LAPORAN PRAKTIKUM STRUKTUR DATA DAN ALGORITMA**

## **MODUL VI STACK**



**Disusun Oleh :**

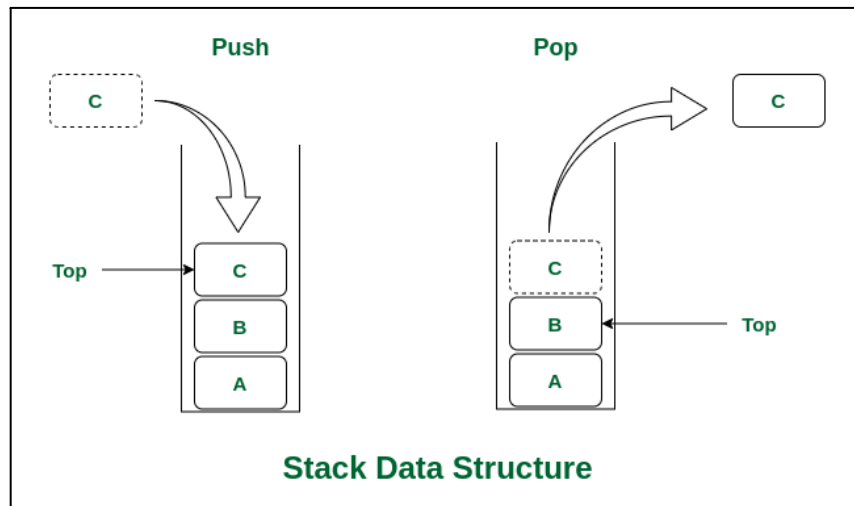
Muhammad Rusdiyanto Asatman  
2311102053

**Dosen :**

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

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

## A. Dasar Teori



Gambar 1.0 Ilustrasi Stack

Stack atau tumpukan adalah struktur data linear yang bekerja berdasarkan prinsip LIFO (Last In First Out), yang artinya data yang terakhir dimasukkan ke dalam stack akan menjadi data yang pertama kali diambil atau dikeluarkan dari stack. Stack memiliki beberapa operasi yang umumnya ada pada stack, di antara lain :

- Push, yaitu menambahkan elemen baru ke dalam stack pada posisi teratas.
- Pop, yaitu menghapus elemen teratas dari stack dan mengembalikan nilainya.
- Peek atau Top, yaitu mengecek elemen teratas dari stack tanpa menghapusnya.
- IsEmpty, yaitu memeriksa apakah stack kosong atau tidak.
- Size, yaitu mengembalikan jumlah elemen dalam stack.

Stack sering digunakan dalam berbagai aplikasi pemrograman. Salah satu contohnya adalah implementasi undo/redo dalam aplikasi text editor.

## B. Guided

### Guided 1

```
#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;
}

```

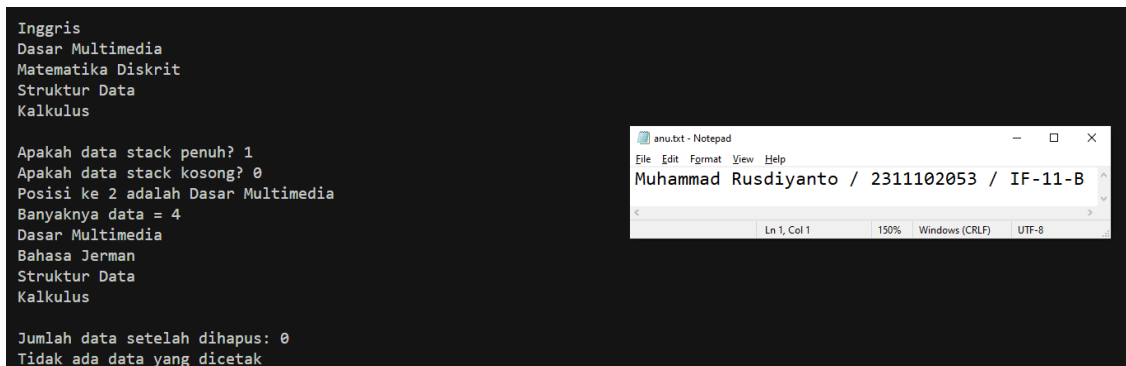
```

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



```
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
```

The screenshot shows a terminal window with a dark background. It displays the output of a program, including a list of subjects (Inggris, Dasar Multimedia, Matematika Diskrit, Struktur Data, Kalkulus) and the results of stack operations (checking if full, checking if empty, pushing data, and popping data). In the background, a Notepad window is visible with the text 'Muhammad Rusdiyanto / 2311102053 / IF-11-B'.

## Deskripsi:

Program di atas adalah program yang mendemonstrasikan penggunaan dari Stack. Stack dalam program ini dapat menyimpan maksimal 5 data dalam bentuk string. Program ini memiliki 2 variabel yang cukup penting yaitu maksimal dan top. Variabel maksimal digunakan untuk menyimpan jumlah data maksimum yang dapat dimasukkan ke dalam array, sementara Variabel top digunakan untuk menyimpan banyak data saat itu. Program ini memiliki beberapa fungsi yang terdiri dari :

1. isFull : Digunakan untuk mengecek apakah Stack sudah penuh atau belum dengan cara membandingkan nilai top dan maksimum. Jika top sama dengan maksimum, maka akan dikembalikan true (artinya sudah penuh). Jika tidak, maka akan dikembalikan nilai false (artinya belum penuh).
2. isEmpty : Digunakan untuk mengecek apakah Stack masih kosong. Jika top sama dengan 0 , maka nilai true akan dikembalikan (artinya masih kosong). Selain 0, maka nilai false akan dikembalikan (artinya sudah ada isinya).
3. pushArrayBuku : Berfungsi untuk menambahkan data ke dalam Stack. Pertama, program akan mengecek apakah Stack sudah penuh atau belum. Jika tidak, maka data akan ditambahkan dan nilai top akan ditambah satu. Jika penuh, maka pesan akan ditampilkan.
4. popArrayBuku : Berfungsi untuk menghapus nilai di posisi paling atas (indeks paling besar, nilai top dikurangi 1). Cara kerja fungsi ini dimulai dengan mengecek apakah Stack masih kosong atau tidak. Jika tidak kosong, maka program akan menghapus nilai paling atas dari Stack, dalam artian lain, program akan menghapus nilai yang terakhir dimasukkan ke Stack. Jika masih kosong, maka program akan menampilkan pesan.

5. peekArrayBuku : Digunakan untuk menampilkan nilai data sesuai dengan posisi yang diminta. Proses dimulai dengan pengecekan apakah Stack kosong atau tidak, sama seperti popArrayBuku. Jika kosong, maka program akan menampilkan pesan. Jika tidak kosong, maka program akan menampilkan data Stack pada posisi yang diminta.
6. countStack : Berfungsi untuk mengembalikan nilai top (sebenarnya tidak perlu memakai countStack, karena bisa langsung memanggil nilai top).
7. changeArrayBuku : Digunakan untuk mengubah nilai di dalam Stack. Hal ini dilakukan dengan cara mengecek data yang ada terlebih dahulu. Jika indeks/posisi data yang dimasukkan lebih besar dari jumlah data yang ada (dalam hal ini menggunakan variabel top), maka program akan menampilkan pesan. Jika tidak, maka program akan melanjutkan proses dan mengganti nilai sesuai dengan indeks yang diinputkan.
8. destroyArrayBuku : Digunakan untuk menghapus semua data yang ada di dalam Stack dengan cara looping ke setiap indeks dalam Stack, dimulai dari data paling atas (top).
9. cetakArrayBuku : Berfungsi untuk menampilkan semua nilai dalam Stack. Hal ini dilakukan dengan cara looping ke setiap indeks dalam stack, dimulai dari data paling atas (top).

## C. Unguided

### Unguided 1

```
#include <iostream>

using namespace std;

string reverse(string sentence) {
    string temp;
    int length = sentence.length();
    for (int i = length; i > 0; i--) temp += sentence[i - 1];
    return temp;
}

int main() {
    string sentence;

    cout << "Masukkan kalimat : ";
    getline(cin, sentence);

    if (sentence.length() == 0) {
        cout << "Kalimat masih kosong.";
        return 0;
    }

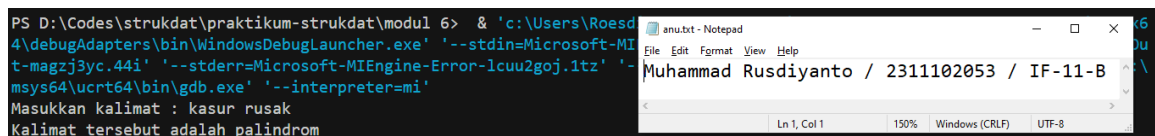
    string reversed = reverse(sentence);

    if (reversed == sentence) {
        cout << "Kalimat tersebut adalah palindrom\n";
    }
    else {
        cout << "Kalimat tersebut bukan palindrom\n";
    }

    return 0;
}
```

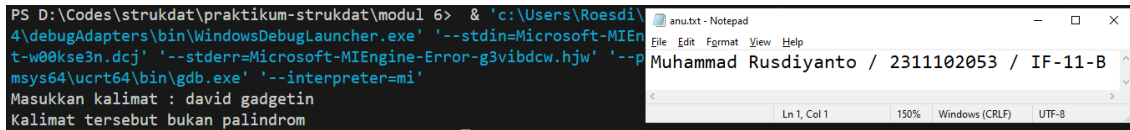
### Screenshots Output

#### # Palindrom





## # Bukan Palindrom



```
PS D:\Codes\strukdat\praktikum-strukdat\modul 6> & 'c:\Users\Roesdi\4\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-t-w0kxe3n.dcj' '--stderr=Microsoft-MIEngine-Error-g3vibdew.hjw' '--pmsys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Masukkan kalimat : david gadgetin
Kalimat tersebut bukan palindrom
```

### Deskripsi:

Program di atas adalah sebuah program yang dapat menentukan apakah suatu kalimat merupakan palindrom atau bukan. Palindrom merupakan suatu kata/kalimat yang tidak berbeda meskipun dibalik, misalkan seperti kata kakak (ketika dibalik menjadi `kakak`, katanya masih sama).

Program ini dimulai dengan meminta kalimat/kata yang akan dicek. Setelah pengguna menginputkan kalimat, program mengecek terlebih dahulu apakah input masih kosong atau tidak. Jika masih kosong, maka program berhenti. Jika tidak, maka program lanjut. Setelah dicek, program baru akan melakukan pembalikan terhadap kata/kalimat dengan memanggil fungsi `reverse`.

Fungsi `reverse` merupakan fungsi yang digunakan untuk membalikkan kata atau kalimat. Proses pembalikan kata dimulai dengan pendeklarasian variabel temp (untuk menyimpan kata yang telah dibalik) dan variabel length (untuk menyimpan panjang kalimat/kata). Kemudian, program akan melakukan looping untuk setiap karakter/huruf dari kata/kalimat, dimulai dari karakter/huruf terakhir sampai karakter/huruf awal. Di sini, looping dapat dilakukan karena tipe data string pada dasarnya adalah array dari char (barisan char atau karakter yang membentuk kata/kalimat). Setelah looping selesai, program akan mengembalikan nilai string yang ditampung dalam variabel temp.

Nilai dari proses pembalikan kalimat/kata akan ditampung dalam variabel `reversed`. Setelah itu, program akan membandingkan kata/kalimat yang ditampung dalam variabel `reversed` dan variabel `sentence`. Jika kedua kalimat/kata sama, maka program akan menampilkan output bahwa kalimat/kata yang dimasukkan adalah palindrom. Jika tidak sama, maka program akan mengoutputkan pesan bahwa kalimat/kata bukan palindrom.

## Unguided 2

```
#include <iostream>
#include <sstream>

using namespace std;

bool hasThreeWords(string sentence) {
    stringstream stream(sentence);
    string temp;
    int count = 0;
    while (stream >> temp) count++;
    if (count >= 3) return true;
    else return false;
}

string reverse(string sentence) {
    string temp;
    int length = sentence.length();
    for (int i = length; i > 0; i--) temp += sentence[i - 1];
    return temp;
}

int main() {
    string sentence;

    cout << "Masukkan kalimat [Min. 3 kata] : ";
    getline(cin, sentence);

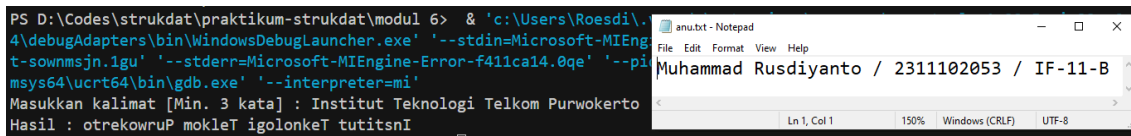
    if (sentence.length() == 0) {
        cout << "Kalimat masih kosong.";
        return 0;
    }
    else if (!hasThreeWords(sentence)) {
        cout << "Kalimat kurang dari 3 kata";
        return 0;
    }

    string reversed = reverse(sentence);
    cout << "Hasil : " << reversed << endl;

    return 0;
}
```

## Screenshots Output

# Lebih dari 3 kata



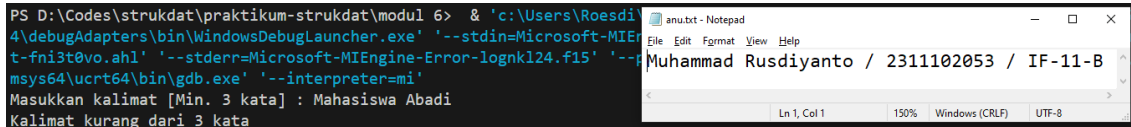
The screenshot shows a Windows command prompt window with the following text:

```
PS D:\Codes\strukdat\praktikum-strukdat\modul 6> & 'c:\Users\Roesdi\4\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-t-sownmsjn.1gu' '--stderr=Microsoft-MIEngine-Error-f411ca14.0qe' '--pi-msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Masukkan kalimat [Min. 3 kata] : Institut Teknologi Telkom Purwokerto
Hasil : otrekowruP mokleT igolonkeT tutitsnI
```

Next to it is a Notepad window titled 'anu.txt - Notepad' containing the text:

Muhammad Rusdiyanto / 2311102053 / IF-11-B

# Kurang dari 3 kata



The screenshot shows a Windows command prompt window with the following text:

```
PS D:\Codes\strukdat\praktikum-strukdat\modul 6> & 'c:\Users\Roesdi\4\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-t-fni3t0vo.ahl' '--stderr=Microsoft-MIEngine-Error-lognk124.f15' '--pi-msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Masukkan kalimat [Min. 3 kata] : Mahasiswa Abadi
Kalimat kurang dari 3 kata
```

Next to it is a Notepad window titled 'anu.txt - Notepad' containing the text:

Muhammad Rusdiyanto / 2311102053 / IF-11-B

## Deskripsi:

Program di atas adalah sebuah program untuk membalikkan kalimat yang dimasukkan. Program tersebut kurang lebih sama dengan program pada bagian Unguided 1, namun dengan beberapa penambahan dan modifikasi.

Alur kerja program ini dimulai dari fungsi main, dimana program meminta pengguna untuk memasukkan kalimat dengan ketentuan minimal 3 kata. Kemudian, program akan mengecek apakah input tadi kosong atau tidak. Jika kosong, maka program akan berhenti. Jika tidak, maka program akan melanjutkan pengecekan lagi dengan memanggil fungsi `hasThreeWords`.

Fungsi `hasThreeWords` dalam program ini merupakan sebuah fungsi yang digunakan untuk mengecek apakah suatu kalimat memiliki 3 atau lebih kata. Dalam fungsi itu, kalimat dicek menggunakan `stringstream` yang secara otomatis dapat melakukan pembacaan string per kata, tanpa memerlukan kode yang banyak. Kata dalam kalimat akan dihitung satu per satu dengan cara looping stream. Jika setelah looping, jumlah kata terhitung sama dengan atau lebih dari 3, maka program akan mengembalikan nilai true. Jika kurang dari 3, maka program akan mengembalikan nilai false.

Program akan melanjutkan proses berdasarkan nilai yang dikembalikan oleh fungsi `hasThreeWords`. Jika dikembalikan nilai true, maka program akan lanjut. Jika dikembalikan nilai false, maka program akan menampilkan pesan dan berhenti. Kalimat yang telah dicek tadi kemudian akan dibalik menggunakan fungsi `reverse`, lalu hasil kalimat yang dibalik akan ditampilkan ke pengguna.

## D. Kesimpulan

Dalam kesimpulannya, stack adalah struktur data linier yang mengikuti prinsip Last In, First Out (LIFO), di mana elemen terakhir yang dimasukkan adalah yang pertama kali dikeluarkan. Dengan operasi-operasi dasarnya seperti push, pop, dan peek, stack sangat berguna dalam situasi-situasi di mana urutan akses data sangat penting. Hal ini sering digunakan dalam implementasi aplikasi, contohnya dalam fitur undo dan redo text editor. Kesederhanaan konsep dan kegunaan yang luas membuat stack menjadi struktur data yang penting dalam pemrograman dan aplikasi komputer secara umum.

## E. Referensi

Asisten Praktikum, "Modul 6 Stack"

Stackoverflow. (2010). "C function to count all the words in a string." Diakses pada 15 Mei 2024, dari <https://stackoverflow.com/questions/3672234/c-function-to-count-all-the-words-in-a-string>.

Alawiah. "Kenalan dengan Pengertian Stack dalam Pemrograman." Diakses pada 16 Mei 2024, dari <https://bakrie.ac.id/articles/552-kenalan-dengan-pengertian-stack-dalam-pemrograman.html>.