

**LAPORAN PRAKTIKUM
STRUKTUR DATA DAN ALGORITMA
MODUL 6
“STACK”**



**DISUSUN OLEH:
TSAQIF KANZ AHMAD
2311102075
IF-11-B**

DOSEN:

WAHYU ANDI SAPUTRA S.Pd., M.Eng.

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

TUJUAN PRAKTIKUM

1. Mampu memahami konsep stack pada struktur data dan algoritma.
2. Mampu mengimplementasikan operasi – operasi pada stack.
3. Mampu memecahkan permasalahan dengan solusi stack.

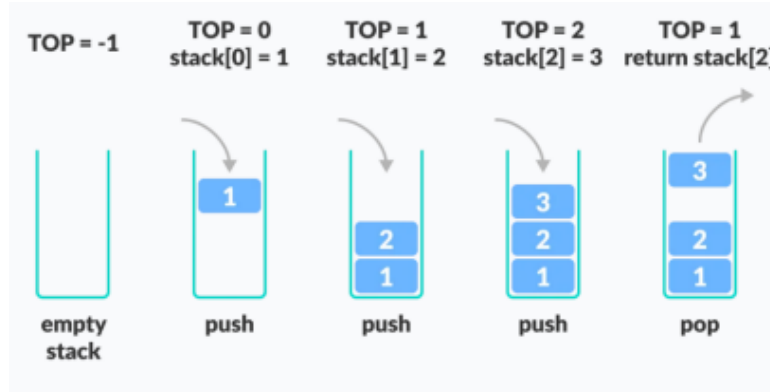
DASAR TEORI

1). PENGERTIAN STACK

Stack merupakan struktur data sederhana yang digunakan untuk menyimpan data (mirip seperti linked list). Dalam sebuah tumpukan, urutan kedatangan data itu penting. Tumpukan piring di kafetaria adalah contoh tumpukan yang bagus. Piring ditambahkan ke tumpukan saat dibersihkan dan ditempatkan di atas. Bila piring dibutuhkan, piring itu diambil dari tumpukan paling atas. Piring pertama yang dimasukkan ke dalam tumpukan adalah piring terakhir yang digunakan. Dalam stack, data disimpan dan diakses menggunakan prinsip LIFO (Last In, First Out), yang berarti data yang terakhir dimasukkan ke dalam stack akan menjadi data yang pertama kali diambil atau dikeluarkan.

Konsep ini mirip dengan tumpukan buku, di mana buku yang terakhir diletakkan di atas tumpukan harus diambil terlebih dahulu sebelum bisa mengakses buku yang berada di bagian bawah tumpukan. Operasi utama pada stack meliputi:

- 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.
- IsFull (Penuh), yaitu Memeriksa apakah tumpukan penuh atau tidak (terutama pada implementasi tumpukan dengan kapasitas terbatas).
- Top (Atas), yaitu Mendapatkan nilai atau melihat elemen teratas pada tumpukan tanpa menghapusnya
- Clear (Hapus Semua), yaitu Mengosongkan atau menghapus semua elemen dari tumpukan.
- Search (Cari): Mencari keberadaan elemen tertentu dalam tumpukan.



Gambar Cara bekerja pada Struktur Data Stack

2). IMPLEMENTASI DAN KEGUNAAN STACK

Stack dapat diimplementasikan menggunakan struktur data array atau linked list. Implementasi dengan array lebih sederhana dan memberikan akses acak kepada pengguna berdasarkan indeks. Namun, ukuran array tidak dapat berubah secara dinamis sesuai dengan kebutuhan. Implementasi dengan linked list lebih fleksibel karena dapat bertambah dan berkurang sesuai dengan kebutuhan saat runtime, namun memerlukan alokasi memori tambahan karena penggunaan pointer dalam linked list. Stack sering digunakan dalam berbagai konteks pemrograman, seperti manajemen memori, pemanggilan fungsi, evaluasi ekspresi matematika, dan manajemen tumpukan panggilan saat melakukan rekursi.

PENJELASAN GUIDED

1). 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;
}

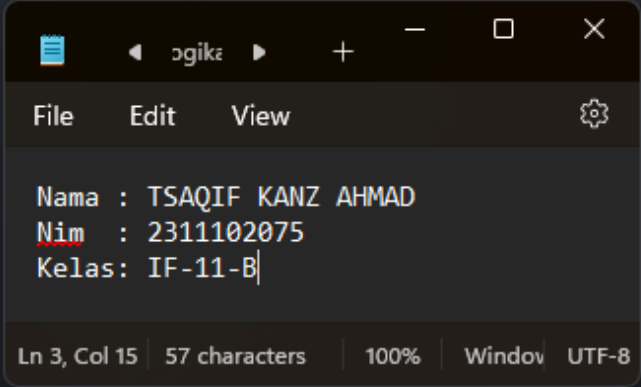
```

SCREENSHOT OUTPUT

```
PS C:\Users\ACER> cd "C:\Users\ACER\AppData\Local\Temp\" ; if ($?) { g++ tempCodeRunnerFile.cpp
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 C:\Users\ACER\AppData\Local\Temp>
```



The screenshot shows a code editor window with a dark theme. The window has a title bar with standard Windows controls (back, forward, zoom in, zoom out, close) and a menu bar with 'File', 'Edit', and 'View'. The main text area displays the following information: 'Nama : TSAQIF KANZ AHMAD', 'Nim : 2311102075', and 'Kelas: IF-11-B'. The status bar at the bottom indicates 'Ln 3, Col 15', '57 characters', '100%', 'Window', and 'UTF-8'.

DESKRIPSI PROGRAM

Kode di atas merupakan implementasi struktur data stack menggunakan array. Pada kode program diatas tersebut terdapat berbagai tipe data beserta fungsinya diantaranya sebagai berikut. Pada fungsi `pushArrayBuku()` digunakan untuk menambahkan 5 buku ke dalam tumpukan. Kemudian, fungsi `cetakArrayBuku()` digunakan untuk menampilkan isi tumpukan. Fungsi `isFull()` dan `isEmpty()` digunakan untuk memeriksa status tumpukan. Fungsi `peekArrayBuku()` digunakan untuk melihat data pada posisi tertentu. Fungsi `popArrayBuku()` digunakan untuk menghapus data terakhir dari tumpukan. Fungsi `countStack()` digunakan untuk menghitung jumlah data di dalam tumpukan. Fungsi `changeArrayBuku()` digunakan untuk mengubah data pada posisi tertentu. Fungsi `destroyArraybuku()` digunakan untuk menghapus semua data dari dalam tumpukan.

PENJELASAN 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. Contoh :

Kalimat : ini

Kalimat tersebut adalah polindrom

Kalimat : telkom

Kalimat tersebut adalah bukan polindrom

```
Masukan Kalimat : ini
Kalimat tersebut adalah : Palindrom
```

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?

Contoh :

Kalimat : Telkom Purwokerto

Hasil : otrekowruP mokleT

```
Masukkan Kata Telkom Purwokerto
Datastack Array :
Data : otrekowruP mokleT
```

1). SOURCE CODE

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

using namespace std;

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

    for (char ch : str) {
        if (isalpha(ch)) {
            cleanedStr += tolower(ch);
        }
    }

    for (char ch : cleanedStr) {
        s.push(ch);
    }

    for (char ch : cleanedStr) {
        if (ch != s.top()) {
            return false;
        }
        s.pop();
    }

    return true;
}

int main() {
    string input;

    cout << "Masukan Kalimat\t\t : ";
    getline(cin, input);

    if (isPalindrome(input)) {
        cout << "Kalimat \"" << input << "\" adalah\t : Palindrom" <<
endl;
    } else {
        cout << "Kalimat \"" << input << "\" adalah : Bukan Palindrom" <<
endl;
    }

    return 0;
}
```


SCREENSHOT OUTPUT

PS C:\Users\ACER> cd "C:\Users\ACER\AppData\Local\Temp"	Nama : TSAQIF KANZ AHMAD
Masukan Kalimat : ini	Nim : 2311102075
Kalimat "ini" adalah : Palindrom	Kelas: IF-11-B
PS C:\Users\ACER\AppData\Local\Temp>	

PS C:\Users\ACER> cd "C:\Users\ACER\AppData\Local\Temp"	Nama : TSAQIF KANZ AHMAD
Masukan Kalimat : Telkom	Nim : 2311102075
Kalimat "Telkom" adalah : Bukan Palindrom	Kelas: IF-11-B
PS C:\Users\ACER\AppData\Local\Temp>	

DESKRIPSI PROGRAM

Program diatas adalah program untuk memeriksa apakah kalimat yang dimasukan merupakan palindrom atau bukan. Pada program tersebut terdapat 3 uraian dari setiap program yaitu Header, fungsi pada tipe 'isPalindrom(const string &str)' dan fungsi main(). Pada bagian Header terdapat tipe data stack, ctype dan string sebagai penanda memulainya suatu data serta memanipulasi pada struktur data tersebut. Pada bagian fungsi 'isPalindrom(const string &str)' untuk menerima sebuah string sebagai parameter referensi dan mengembalikan nilai true jika string tersebut merupakan palindrom, Jika false maka bukan. Pada fungsi 'main()' sebagai fungsi utama program. Pada Deklarasi string input untuk menyimpan kalimat yang dimasukan user, mencetak pesan meminta pengguna memasukan kalimat dan membaca kalimat yang dimasukan pengguna menggunakan 'getline(cin,input)'. Pada pemanggilan fungsi 'isPalindrom(input)' untuk memeriksa apakah kalimat tersebut palidrom atau bukan.

2). SOURCE CODE

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

using namespace std;

string reverseWords(const string &str) {
    stack<char> s;
    string result = "";

    //Membalikkan setiap karakter dalam string menggunakan stack
    for (char ch : str) {
        s.push(ch);
    }

    //Mengambil karakter dari stack untuk membentuk kalimat yang dibalik
    while (!s.empty()) {
        result += s.top();
        s.pop();
    }

    return result;
}

int main() {
    string input;

    cout << "Masukkan kalimat (minimal 3 kata): ";
    getline(cin, input);

    //Memastikan bahwa input memiliki minimal 3 kata
    int wordCount = 0;
    for (char ch : input) {
        if (ch == ' ') {
            wordCount++;
        }
    }
    wordCount++; //Menambah satu untuk kata terakhir

    if (wordCount < 3) {
        cout << "Error: Kalimat harus memiliki minimal 3 kata." << endl;
    } else {
        string reversed = reverseWords(input);
        cout << "Hasil: " << reversed << endl;
    }

    return 0;
}
```

SCREENSHOT OUTPUT

```
PS C:\Users\ACER> cd "C:\Users\ACER\AppData\Local\Temp"
Masukkan kalimat (minimal 3 kata): Telkom Un
Hasil: otrekowruP ytisrevinU mokleT
PS C:\Users\ACER\AppData\Local\Temp> 
```

Nama : TSAQIF KANZ AHMAD
Nim : 2311102075
Kelas: IF-11-B

DESKRIPSI PROGRAM

Program diatas adalah program untuk membalikan kata-kata dalam kalimat yang dimasukan. Pada bagian Header terdapat tipe data stack dan string untuk menggunakan struktur data stack (penumpukan) dan untuk manipulasi string. Pada fungsi 'reverseWords(const string &str)' untuk mengembalikan string baru dengan kata-kata yang dibalik. Dan fungsi 'main()' memastikan bahwa input memiliki minimal 3 kata, membalikkan kata-kata menggunakan 'reverseWords', dan menampilkan hasil kalimat yang dibalik pada program tersebut.

KESIMPULAN

Stack adalah struktur data linier yang mengikuti aturan Last In, First Out (LIFO). Artinya, elemen yang terakhir dimasukkan ke dalam stack adalah elemen yang pertama kali dikeluarkan. Stack sering dianalogikan dengan tumpukan piring, di mana piring yang paling atas adalah piring yang paling terakhir diletakkan dan piring yang paling atas itulah yang pertama kali diambil. Struktur data pada Stack juga digunakan dalam berbagai aplikasi dalam ilmu computer seperti web, kalkulator, Mesin virtual dsb. Pemahaman tentang stack sangat penting untuk mempelajari algoritma dan struktur data yang lebih kompleks supaya pengguna bisa memahami serta menerapkan stack dalam kehidupan sehari-hari dalam membuat aplikasi dalam komputer.

DAFTAR PUSTAKA

- [1] Asisten Praktikum, "Modul 6 STACK" : [MODUL 6 STACK.pdf](#)
- [2] Pengertian Stack dalam Pemrograman – Universitas Bakrie:
[https://bakrie.ac.id/articles/552-kenalan-dengan-pengertian-stack dalam-pemrograman.html#:~:text=Stack%20merupakan%20sebuah%20struktur%20data,Last%20In%2C%20First%20Out\).](https://bakrie.ac.id/articles/552-kenalan-dengan-pengertian-stack-dalam-pemrograman.html#:~:text=Stack%20merupakan%20sebuah%20struktur%20data,Last%20In%2C%20First%20Out).)