

**LAPORAN PRAKTIKUM
STRUKTUR DATA DAN ALGORITMA**

MODUL 6



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

Disusun oleh:

ABDA FIRAS RAHMAN

2311102049

IF-11-B

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

BAB I

DASAR TEORI

STACK

1. Pengertian

Stack merupakan sebuah struktur data yang digunakan untuk menyimpan sejumlah objek atau variabel. Sesuai dengan namanya, stack sering kali terlihat seperti tumpukan dalam bentuk fisik. Karakteristik khas dari stack adalah penggunaan aturan LIFO (Last In, First Out). Artinya, data yang terakhir dimasukkan ke dalam stack akan menjadi data pertama yang diambil atau dikeluarkan.

Cara struktur data stack dalam menyimpan sebuah nilai dapat kita bayangkan seperti piring yang disusun rapi secara bertumpuk ke atas. Apabila kita ingin mengambil piring bagian bawah, kita harus terlebih dahulu menyisihkan semua piring yang ada di atas. Dalam istilah pemrograman, upaya menambahkan elemen pada struktur data stack disebut dengan push. Sedangkan proses menghapus atau menghilangkan elemen data dari stack disebut pop.

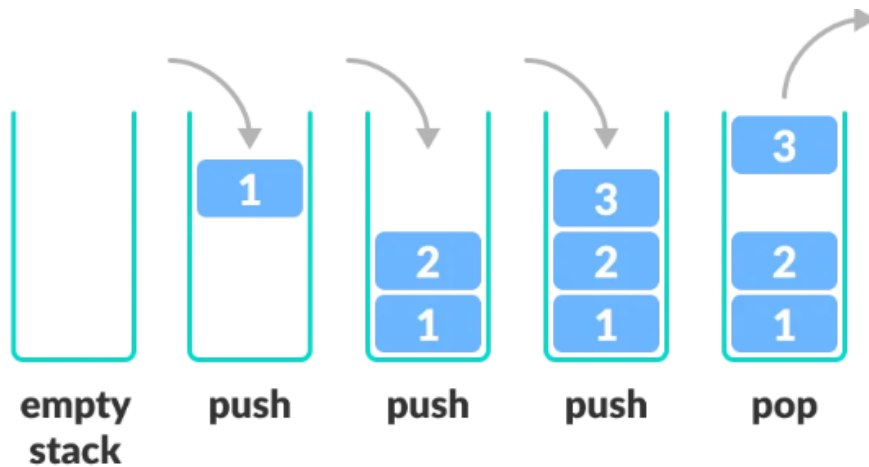
Konsep ini mirip dengan tumpukan buku, di mana buku yang terakhir diletakkan di atas tumpukan harus diambil terlebih dahulu sebelum bisa mengakses buku yang berada di bagian bawah tumpukan.

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), Memeriksa apakah tumpukan penuh atau tidak (terutama pada implementasi tumpukan dengan kapasitas terbatas)
- Clear (Hapus Semua), Mengosongkan atau menghapus semua

elemen dari tumpukan.

- Search (Cari): Mencari keberadaan elemen tertentu dalam tumpukan.



Sumber: programiz.com

Dari gambar di atas, dapat terlihat bahwa meskipun elemen ke-3 adalah yang paling terakhir ditambahkan, namun elemen tersebut justru yang pertama dihapus. Operasi inilah yang kemudian disebut sebagai prinsip operasi LIFO (Last In First Out).

2. GUIDED

```
#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";
}

```

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
```

File Edit View	
NAMA	: ABDA FIRAS RAHMAN
NIM	: 2311102049
KELAS	: IF-11-B

Ln 3, Col 16 | 57 characters | 100% | Window

DESKRIPSI PROGRAM

Program ini adalah implementasi struktur data stack (tumpukan) menggunakan array di C++. Stack digunakan untuk menyimpan data secara sementara dengan metode LIFO (Last In First Out), di mana data yang terakhir masuk akan menjadi data yang pertama keluar.

Mengimplementasikan stack menggunakan array dengan kapasitas maksimal lima elemen. Fungsi `isFull` dan `isEmpty` memeriksa apakah stack penuh atau kosong. Fungsi `pushArrayBuku` menambahkan elemen ke stack jika belum penuh, sedangkan `popArrayBuku` menghapus elemen teratas jika tidak kosong. Fungsi `peekArrayBuku` menampilkan elemen pada posisi tertentu dari atas stack, dan `countStack` mengembalikan jumlah elemen dalam stack.

Fungsi `changeArrayBuku` mengubah elemen pada posisi tertentu, dan `destroyArraybuku` mengosongkan seluruh stack. Fungsi `cetakArrayBuku` mencetak seluruh elemen stack dari atas ke bawah. Dalam fungsi `main`, program menambahkan lima buku, mencetak stack, memeriksa apakah stack penuh atau kosong, menampilkan elemen pada posisi tertentu, menghapus elemen teratas, mencetak jumlah elemen, mengubah elemen, dan mencetak kembali stack setelah perubahan.

UNGUIDED (1)

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

using namespace std;

// Fungsi untuk menghapus karakter non-alfabet dari string
string removeNonAlphanumeric(string str) {
    string result = "";
    for (char c : str) {
        if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z')) {
            result += tolower(c); // Ubah huruf menjadi lowercase
        }
    }
    return result;
}

// Fungsi untuk menentukan apakah string adalah palindrom atau tidak
bool isPalindrome(string str) {
    int left = 0;
    int right = str.length() - 1;

    while (left < right) {
        // Lewati karakter non-alfabet di kiri
        while (left < right && !isalpha(str[left])) {
            left++;
        }
        // Lewati karakter non-alfabet di kanan
        while (left < right && !isalpha(str[right])) {
            right--;
        }
        // Periksa apakah karakter di kiri sama dengan karakter di kanan
        if (tolower(str[left]) != tolower(str[right])) {
            return false;
        }
        left++;
        right--;
    }

    return true;
}

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

    string cleanedString = removeNonAlphanumeric(kalimat);
```

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

return 0;
}
```

OUTPUT

```
Masukkan kalimat: ini
Kalimat tersebut adalah palindrom.
PS C:\Users\LENOVO\OneDrive\Documents\KTIKUM STRUKTUR DATA SMT 2\modul 6 pra
S\MATA KULIAH SEMESTER 2\STRUKTUR DATA unguided1.cpp -o unguided1 } ; if ($?
Masukkan kalimat: telkom
Kalimat tersebut bukan palindrom.
PS C:\Users\LENOVO\OneDrive\Documents\DOKUMEN FIRAS KULIAH\DOKUMEN FIR
KTIKUM STRUKTUR DATA SMT 2\modul 6 praktikum strukdat> 
```

File	Edit	View
NAMA : ABDA FIRAS RAHMAN		
KELAS : IF-11-B		
NIM : 2311102049		
Ln 3, Col 17	57 characters	100%

DESKRIPSI PROGRAM

Program tersebut merupakan sebuah program C++ yang berfungsi untuk memeriksa apakah sebuah kalimat yang dimasukkan pengguna merupakan palindrom atau bukan. Palindrom adalah sebuah kata, frasa, angka, atau susunan lainnya yang dapat dibaca dengan cara yang sama dari depan maupun belakang.

Dimulai dengan memasukkan pustaka iostream dan string yang diperlukan untuk operasi input/output dan manipulasi string, serta menggunakan namespace std agar penulisan kode lebih ringkas.

Program ini memiliki dua fungsi utama. Fungsi pertama, removeNonAlphanumeric, berfungsi untuk menghapus karakter non-alfabet dari sebuah string dan mengubah semua huruf menjadi huruf kecil. Fungsi ini menggunakan loop untuk memeriksa setiap karakter dalam string; jika karakter tersebut merupakan huruf, karakter tersebut ditambahkan ke string hasil setelah diubah menjadi huruf kecil.

Fungsi kedua, `isPalindrome`, digunakan untuk memeriksa apakah string yang diberikan adalah palindrom. Fungsi ini bekerja dengan membandingkan karakter dari dua arah (kiri dan kanan). Ia melewati karakter non-alfabet dan memeriksa apakah karakter yang berlawanan adalah sama, dengan mengabaikan perbedaan huruf besar dan kecil. Jika ditemukan ketidaksesuaian, fungsi ini segera mengembalikan nilai `false`, menandakan bahwa string tersebut bukan palindrom.

Fungsi `main` menggabungkan kedua fungsi ini untuk memproses input pengguna. Program meminta pengguna memasukkan sebuah kalimat, yang kemudian dibersihkan dari karakter non-alfabet menggunakan `removeNonAlphanumeric`. Hasil pembersihan ini kemudian diperiksa apakah merupakan palindrom menggunakan `isPalindrome`. Hasil pemeriksaan ini dicetak ke layar sebagai output, menunjukkan apakah kalimat yang dimasukkan adalah palindrom atau bukan. Dengan demikian, program ini secara efektif memungkinkan pengguna untuk mengecek palindrom secara interaktif dengan memperhitungkan hanya karakter alfabet dan mengabaikan perbedaan kapitalisasi serta karakter non-alfabet.

UNGUIDED (2)

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

using namespace std;

// Fungsi untuk membalikkan kalimat menggunakan stack
string reverseSentence(string sentence) {
    stack<char> charStack;
    string reversedSentence = "";

    // Push setiap karakter ke dalam stack
    for (char c : sentence) {
        charStack.push(c);
    }

    // Pop setiap karakter dari stack untuk mendapatkan kalimat terbalik
    while (!charStack.empty()) {
```

```

        reversedSentence += charStack.top();
        charStack.pop();
    }

    return reversedSentence;
}

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

    string hasilPembalikan = reverseSentence(kalimat);
    cout << "Hasil pembalikan: " << hasilPembalikan << endl;

    return 0;
}

```

OUTPUT

<pre> PRATIUM STRUKTUR DATA SMT 2\modul 6 praktikum s S\MATA KULIAH SEMESTER 2\STRUKTUR DATA DAN ALG unguided2.cpp -o unguided2 } ; if (\$?) { .\un Masukkan kalimat atau kata: Telkom Purwokerto Hasil dari pembalikan nya: otrekowruP mokleT PS C:\Users\LENOVO\OneDrive\Documents\DOKUMEN </pre>	<pre> NAMA : ABDA FIRAS RAHMAN KELAS : IF-11-B NIM : 2311102049 </pre>
---	---

DESKRIPSI PROGRAM

Program C++ di atas berfungsi untuk membalikkan urutan karakter dalam sebuah kalimat atau kata yang dimasukkan oleh pengguna. Pertama, program menyertakan pustaka <iostream>, <stack>, dan <string> untuk operasi input/output, penggunaan struktur data stack, dan manipulasi string. Namespace std digunakan untuk mempermudah penulisan kode.

Inti dari program ini adalah fungsi reverseSentence yang menerima sebuah string sebagai input dan mengembalikan string tersebut dalam urutan terbalik. Fungsi ini bekerja dengan membuat sebuah stack untuk menyimpan karakter-karakter dari string. Setiap karakter dalam string dimasukkan ke dalam stack menggunakan operasi push. Setelah semua karakter dimasukkan, karakter-karakter tersebut dikeluarkan dari stack satu per satu menggunakan operasi pop, dan ditambahkan ke string reversedSentence yang akan menjadi hasil akhir.

Fungsi main adalah titik masuk utama program. Dalam fungsi ini, program meminta pengguna untuk memasukkan sebuah kalimat atau kata, lalu membaca input tersebut menggunakan `getline(cin, kalimat)` agar dapat menangkap seluruh kalimat termasuk spasi. Kemudian, program memanggil fungsi `reverseSentence` dengan input dari pengguna dan menyimpan hasilnya dalam variabel `hasilPembalikan`. Akhirnya, hasil pembalikan tersebut ditampilkan kepada pengguna.

DAFTAR PUSTAKA

- 1) Struktur Data Stack: Pengertian, Karakteristik, dan Kegunaannya
<https://www.trivusi.web.id/2022/07/struktur-data-stack.html>
- 2) Kenalan dengan Pengertian Stack dalam Pemrograman!
<https://bakrie.ac.id/articles/552-kenalan-dengan-pengertian-stack-dalam-pemrograman.html>

