

**LAPORAN PRAKTIKUM
STRUKTUR DATA DAN
ALGORITMA**

**MODUL VI
STACK**



Disusun Oleh :

Raka Andriy Shevchenko
2311102054

Dosen

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

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

A. Dasar Teori

Implementasi stack dalam C++ melibatkan pendefinisian metode untuk mendorong dan memunculkan elemen, memeriksa apakah stack kosong atau penuh, dan mempertahankan urutan elemen berdasarkan prinsip Last In First Out (LIFO). Berikut ini adalah gambaran umum yang komprehensif tentang struktur data stack dan implementasinya dalam C++:

Struktur data tumpukan biasanya diimplementasikan menggunakan array dinamis atau daftar taut. Pendekatan array dinamis lebih efisien untuk tumpukan besar karena memungkinkan penyisipan dan penghapusan elemen yang lebih cepat di bagian atas. Pendekatan senarai berantai lebih fleksibel, memungkinkan alokasi memori yang dinamis dan penanganan kebocoran memori yang lebih mudah.

Kompleksitas waktu dari operasi stack tergantung pada implementasinya. Untuk implementasi larik dinamis, operasi push membutuhkan waktu $O(1)$ karena hanya melibatkan pembaruan indeks teratas dan memeriksa apakah tumpukan sudah penuh. Operasi pop juga membutuhkan waktu $O(1)$ karena hanya melibatkan pembaruan indeks teratas dan mengembalikan nilai pada indeks tersebut. Untuk implementasi senarai berantai, operasi push membutuhkan waktu $O(1)$ karena hanya melibatkan pembaruan penunjuk teratas dan membuat simpul baru. Operasi pop membutuhkan waktu $O(1)$ karena hanya melibatkan pembaruan penunjuk teratas dan pengembalian nilai pada simpul teratas.

Kesimpulannya, struktur data stack adalah konsep mendasar dalam ilmu komputer dan banyak digunakan dalam berbagai aplikasi. Implementasi stack dalam C++ melibatkan pendefinisian metode untuk mendorong dan memunculkan elemen, memeriksa apakah stack kosong atau penuh, dan mempertahankan urutan elemen berdasarkan prinsip LIFO. Implementasi larik dinamis lebih efisien untuk stack yang besar, sedangkan implementasi senarai berantai lebih fleksibel dan lebih mudah menangani kebocoran memori.

B. Guided

a. 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";

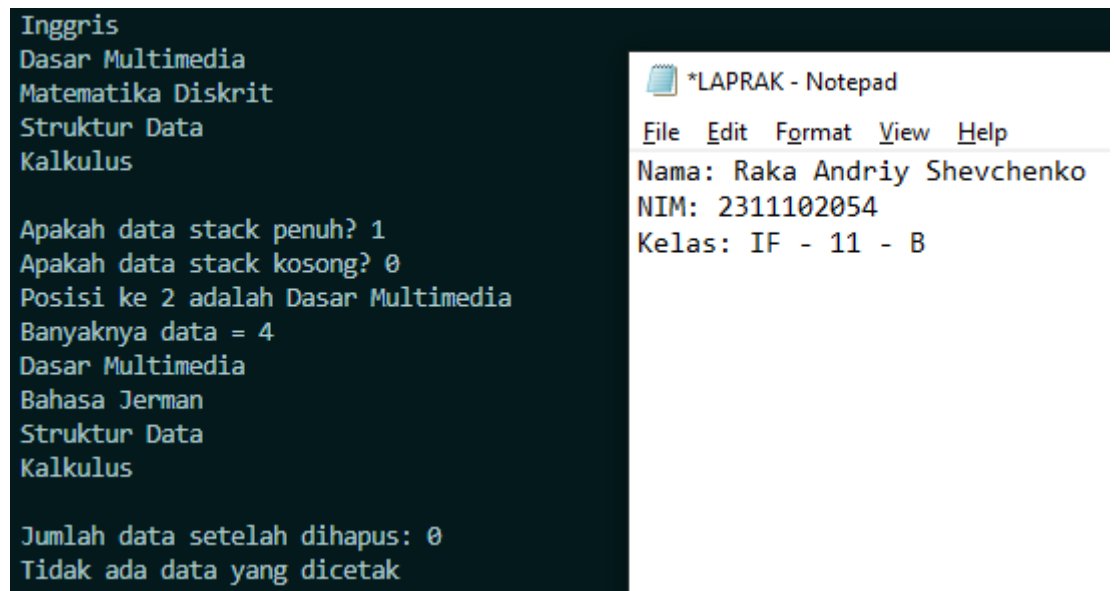
    destroyArraybuku();

    cout << "Jumlah data setelah dihapus: " << top << endl;
    cetakArrayBuku();

    return 0;
}

```

Output:



The screenshot shows the output of a C++ program on the left and a Notepad window on the right. The output text is as follows:

```

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 Notepad window on the right is titled "*LAPRAK - Notepad" and contains the following text:

```

File Edit Format View Help
Nama: Raka Andriy Shevchenko
NIM: 2311102054
Kelas: IF - 11 - B

```

Deskripsi:

Program ini mengimplementasikan sebuah stack menggunakan array untuk menyimpan data buku. Stack adalah struktur data yang mengikuti prinsip LIFO (Last In, First Out). Program ini menyediakan berbagai operasi dasar pada stack, seperti menambah data (push), menghapus data (pop), melihat data (peek), menghitung jumlah data (count), mengubah data (change), menghapus semua data (destroy), dan mencetak data (print).

C. Unguided

a. Unguided 1

Source code:

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

// Fungsi untuk mengubah semua karakter dalam kata menjadi huruf
kecil
string cleankata(const string &kata) {
    string cleaned;
    for (char c : kata) {
        cleaned += tolower(c);
    }
    return cleaned;
}

// Fungsi untuk menentukan apakah kata adalah palindrom
bool isPalindrome(const string &kata) {
    string cleaned_kata = cleankata(kata);
    int length = cleaned_kata.length();

    // Buat stack dengan kapasitas sebesar panjang kata yang
    dibersihkan
    char* stack = new char[length];
    int top = -1;

    // Masukkan semua karakter ke dalam stack
    for (int i = 0; i < length; i++) {
        stack[++top] = cleaned_kata[i];
    }

    string reversed_kata;

    // Buat kata terbalik dengan mengambil karakter dari stack
    while (top != -1) {
        reversed_kata += stack[top--];
    }

    // Hapus stack
```

```

        delete[] stack;

        // Periksa apakah kata asli yang dibersihkan sama dengan
        kata terbalik
        return cleaned_kata == reversed_kata;
    }

int main() {
    string kata;
    cout << "\nMasukkan kata: ";
    cin >> kata;

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

    return 0;
}

```

Output:

```

Masukkan kata: katak
Kata tersebut adalah palindrom.
PS C:\kuliah\Struktur Data\TUGAS 6> ^C
PS C:\kuliah\Struktur Data\TUGAS 6>
PS C:\kuliah\Struktur Data\TUGAS 6> &
IEngine-Out-hwoz4zhh.py2' '--stderr=Mic
IEngine-Pid-mxsvxwgl.mie' '--dbgExe=C:\
Masukkan kata: kata
Kata tersebut bukan palindrom.
PS C:\kuliah\Struktur Data\TUGAS 6> 

```

*LAPRAK - Notepad

File Edit Format View Help

Nama: Raka Andriy Shevchenko
NIM: 2311102054
Kelas: IF - 11 - B

Deskripsi:

Program ini dirancang untuk membaca sebuah kata dari input pengguna, mengubah semua karakternya menjadi huruf kecil dengan fungsi tolower, dan kemudian memeriksa apakah kata tersebut merupakan palindrom dengan menggunakan stack untuk membalik urutan karakter. Setelah membalik urutan karakter, dimasukan hasil tersebut kedalam string reversed kata. Jika kata tersebut sama dengan kata terbalik, maka kata tersebut adalah palindrom.

b. Unguided 2

Source Code:

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

// Fungsi untuk menentukan apakah kalimat adalah palindrom
string terbalik(const string &sentence) {
    string kalimat = sentence;
    int length = kalimat.length();

    // Buat stack dengan kapasitas sebesar panjang kalimat yang
    // dibersihkan
    char* stack = new char[length];
    int top = -1;

    // Masukkan semua karakter ke dalam stack
    for (char c : kalimat) {
        stack[++top] = c;
    }

    string reversed_sentence;

    // Buat kalimat terbalik dengan mengambil karakter dari
    // stack
    while (top != -1) {
        reversed_sentence += stack[top--];
    }

    // Hapus stack
    delete[] stack;

    // Periksa apakah kalimat asli yang dibersihkan sama dengan
    // kalimat terbalik
    return reversed_sentence;
}

int main() {
    string sentence;
    cout << "\nMasukkan kalimat: ";
    getline(cin, sentence);
```

```

    string reversed_sentence = terbalik(sentence);
    cout << "Kalimat terbalik: " << reversed_sentence << endl;

    return 0;
}

```

Output:

<pre> Masukkan kalimat: IT Telkom Purwokerto Kalimat terbalik: otrekowruP mokleT TI PS C:\kuliah\Struktur Data\TUGAS 6> </pre>	<pre> File Edit Format View Help Nama: Raka Andriy Shevchenko NIM: 2311102054 Kelas: IF - 11 - B </pre>
--	---

Deskripsi:

Program ini tidak jauh berbeda dari program pada Unguided 1, program ini membaca sebuah kalimat dari input pengguna, kemudian membalikkan urutan karakternya menggunakan stack, dan menampilkan hasilnya. Bedanya, pada program ini tidak menggunakan fungsi tolower dan menampilkan isi dari reversed_sentence (pada Unguided 1 tidak menampilkan reversed_sentence).

D. Kesimpulan

Pemahaman tentang stack dan implementasinya dalam program sangat penting dalam ilmu komputer dan pemrograman. Stack adalah struktur data dasar yang sering digunakan dalam berbagai algoritma dan aplikasi, seperti analisis ungkapan aritmatika, navigasi mundur (backtracking), dan banyak lagi. Program-program di atas menunjukkan bagaimana stack dapat digunakan secara efektif untuk membalikkan string dan memeriksa polindrom.

E. Referensi

- [1] Daniel Diaz, How to Implement Stack in C++? Amortized analysis. Diakses dari
<http://fajarbaskoro.blogspot.com/2021/06/hash-table.html>
- [2] Sruthy, Stack Data Structure In C++ With Illustration. Diakses dari
http://aren.cs.ui.ac.id/sda/resources/sda2010/15_hashtable.pdf
- [3] Ankith Reddy, C++ Program to Implement Stack using array. Diakses dari
[C++ Program to Implement Stack using array \(tutorialspoint.com\)](http://tutorialspoint.com/cpp-program-to-implement-stack-using-array/)