

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL VI
STACK**



Disusun Oleh :

NAMA : RIFKY DWI MAHARDIKA

NIM : 2311102043

Dosen

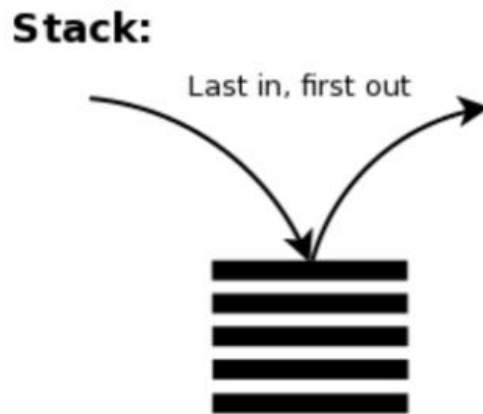
Wahyu Andi Saputra, S.Pd., M.Eng

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

A. Dasar Teori

STACK

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.



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.

Stack sering digunakan dalam berbagai konteks pemrograman, seperti manajemen memori, pemanggilan fungsi, evaluasi ekspresi matematika, dan manajemen tumpukan panggilan saat melakukan rekursi.

Berikut kelebihan dan kekurangan dari stack, antara lain:

- **Kelebihan**

- a. membantu mengelola data dengan metode LIFO
- b. secara otomatis membersihkan objek
- c. tidak mudah rusak
- d. ukuran variabel tidak dapat diubah
- e. mengontrol memori secara mandiri

- **Kekurangan**

- a. memori stack sangat terbatas
- b. ada kemungkinan stack akan meluap atau overflow jika terlalu banyak objek
- c. tidak memungkinkan akses acak, karena harus mengeluarkan tumpukan paling atas terlebih dahulu untuk mengakses tumpukan paling bawah

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

```

PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 6> cd ..
ak 6\ ; if ($?) { g++ guided1.cpp -o guided1 } ; if ($?) { .\guided1 }
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\ASUS\Documents\semester 2\struktur data algoritma\laprak 6>

```

+

Nama : Rifky Dwi Mahardika
NIM : 2311102043

Deskripsi

Program tersebut implementasi dari stack menggunakan array untuk menyimpan data buku. Stack ini memiliki ukuran maksimum lima elemen. Pada pemograman ini menyediakan berbagai fungsi untuk mengelola stack, termasuk menambah data, menghapus data, melihat data tertentu, mengubah data, menghitung jumlah elemen, dan mengosongkan stack. Array Buku: “arrayBuku[5]” menyimpan data buku, dengan variabel “top” sebagai penunjuk elemen teratas dan “maksimal” sebagai kapasitas stack. “isFull()” dan “isEmpty()”: Memeriksa apakah stack penuh atau kosong. pushArrayBuku, menambahkan buku ke stack jika belum penuh. popArrayBuku, menghapus buku dari stack jika tidak kosong. peekArrayBuku“int posisi”, menampilkan buku pada posisi tertentu dari atas stack. countStack(), mengembalikan jumlah buku dalam stack. changeArrayBuku(int posisi, string data), mengubah buku pada posisi tertentu dalam stack. destroyArraybuku(), mengosongkan seluruh stack. cetakArrayBuku():, mencetak semua buku dalam stack dari atas ke bawah.

C. UNGUIDED

UNGUIDED 1

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

char charArray[100];
int maksimal = 100, top = 0;

bool isFull()
{
    return (top == maksimal);
}

bool isEmpty()
{
    return (top == 0);
}

void pushChar(char data)
{
    if (isFull())
    {
        cout << "Stack penuh" << endl;
    }
    else
    {
        charArray[top] = data;
        top++;
    }
}

void popChar()
{
    if (isEmpty())
    {
        cout << "Stack kosong" << endl;
    }
    else
    {
        top--;
    }
}
```



```

char peekChar()
{
    if (!isEmpty())
    {
        return charArray[top - 1];
    }
    else
    {
        return '\0';
    }
}

int main()
{
    string input;
    cout << "=====\n";
    cout << "Masukkan Kalimat: ";
    cin >> input;

    int lengthInput = input.length();
    for (int i = 0; i < lengthInput; ++i)
    {
        pushChar(input[i]);
    }
    bool palindrom = true;
    for (int i = 0; i < lengthInput; ++i)
    {
        if (input[i] != peekChar())
        {
            palindrom = false;
            break;
        }
        popChar();
    }
    if (palindrom)
    {
        cout << "Kalimat tersebut adalah palindrom" << endl;
        cout << "=====\n";
    }
    else
    {
        cout << "Kalimat tersebut bukan palindrom" << endl;
        cout << "=====\n";
    }
    return 0;
}

```

Screenshots Output

- **Contoh kalimat bukan palindrom**

```
PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 6\" ; if ($?) { g++ unguided1.cpp -o unguided1 } ; if ($?) { .\unguided1.exe
=====
Masukkan Kalimat: indonesia
Kalimat tersebut bukan palindrom
=====
PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 6\" : if ($?) { g++ unguided1.cpp -o unguided1 } ; if ($?) { .\unguided1.exe
=====
```

+

Nama : Rifky Dwi Mahardika
NIM : 2311102043

- **Contoh kalimat palindrom**

```
PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 6\" ; if ($?) { g++ unguided1.cpp -o unguided1 } ; if ($?) { .\unguided1.exe
=====
Masukkan Kalimat: radar
Kalimat tersebut adalah palindrom
=====
PS C:\Users\ASUS\Documents\semester 2\struktur data algoritma\laprak 6\" : if ($?) { g++ unguided1.cpp -o unguided1 } ; if ($?) { .\unguided1.exe
=====
```

+

Nama : Rifky Dwi Mahardika
NIM : 2311102043

Deskripsi

Program tersebut adalah implementasi untuk memeriksa apakah sebuah kalimat merupakan palindrom menggunakan stack. Pertama, kalimat dari input user dimasukkan ke dalam stack karakter demi karakter. Kemudian, program membandingkan karakter dari depan kalimat dengan karakter yang diambil dari stack (dari belakang kalimat) satu per satu. Jika semua pasangan karakter sesuai, program menyatakan “kalimat termasuk palindrom”, jika ada pasangan yang tidak sesuai, program menyatakan “kalimat bukan palindrom”. Fungsi-fungsi yang digunakan termasuk untuk memeriksa apakah stack penuh atau kosong, menambahkan dan menghapus karakter dari stack, serta mengambil karakter teratas stack tanpa menghapusnya.

UNGUIDED 2

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

char charArray[100];
int maksimal = 100, top = 0;

bool isFull()
{
    return (top == maksimal);
}

bool isEmpty()
{
    return (top == 0);
}

void pushChar(char data)
{
    if (isFull())
    {
        cout << "Stack penuh" << endl;
    }
    else
    {
        charArray[top] = data;
        top++;
    }
}

void popChar()
{
    if (isEmpty())
    {
        cout << "Stack kosong" << endl;
    }
    else
    {
        charArray[top - 1] = '\0';
        top--;
    }
}
```

```

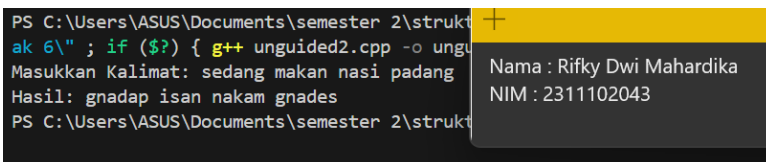
char peekChar()
{
    if (!isEmpty())
    {
        return charArray[top - 1];
    }
    else
    {
        return '\0';
    }
}

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

    int lengthInput = input.length();
    for (int i = 0; i < lengthInput; ++i)
    {
        pushChar(input[i]);
    }
    cout << "Hasil: ";
    while (!isEmpty())
    {
        cout << peekChar();
        popChar();
    }
    return 0;
}

```

Screenshots Output



```

PS C:\Users\ASUS\Documents\semester 2\struktur> g++ unguided2.cpp -o unguided2
ak 6\" ; if ($?) { g++ unguided2.cpp -o unguided2
Masukkan Kalimat: sedang makan nasi padang
Hasil: gnadap isan nakam gnades
PS C:\Users\ASUS\Documents\semester 2\struktur>

```

Deskripsi

Program ini implementasi sederhana dari stack. Program akan membaca sebuah kalimat dari inputan user dan menampilkan kalimat tersebut dalam urutan terbalik menggunakan stack. Program ini memungkinkan user untuk memasukkan sebuah kalimat,

kemudian setiap karakter dari kalimat tersebut dimasukkan ke dalam stack menggunakan fungsi “pushChar”. Setelah itu, program mengeluarkan karakter dari stack satu per satu menggunakan fungsi “peekChar” dan “popChar”, sehingga menghasilkan kalimat yang terbalik sebagai output. Fungsi-fungsi lainnya seperti “isFull” dan “isEmpty” digunakan untuk memeriksa apakah stack penuh atau kosong.

D. Kesimpulan

Stack adalah struktur data linear yang mengikuti prinsip LIFO (Last In, First Out), artinya elemen yang terakhir dimasukkan akan menjadi elemen pertama yang dikeluarkan. Terdapat beberapa operasi utama pada stack, yaitu push (untuk menambahkan elemen ke atas stack), pop (untuk menghapus elemen teratas dari stack), dan peek (untuk melihat elemen teratas tanpa menghapusnya). Stack dapat diimplementasikan menggunakan array atau linked list. Implementasi dengan array memiliki keterbatasan pada ukuran, sedangkan implementasi dengan linked list lebih fleksibel namun memerlukan alokasi memori tambahan. Dengan pemahaman tentang konsep dasar, operasi utama, implementasi, dan penggunaan stack, kita dapat memahami bagaimana menggunakan dan menerapkan struktur data stack dalam pemrograman untuk menyelesaikan berbagai masalah.

E. Referensi

[1] Asisten Praktikum. (2024). Modul VI : Stack

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

[3] Soffyan Ranti, 01 Des 2022, Pengertian Stack dan Queue serta Contoh Penerapannya, Diakses pada 20 Mei 2024, dari <https://tekno.kompas.com/read/2022/12/01/02150047/pengertian-stack-dan-queue-serta-contoh-penerapannya>