

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL VI
STACK**



Disusun Oleh :

NAMA : Fadhel Yussie Ramadhan

NIM : 2311102322

Dosen

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

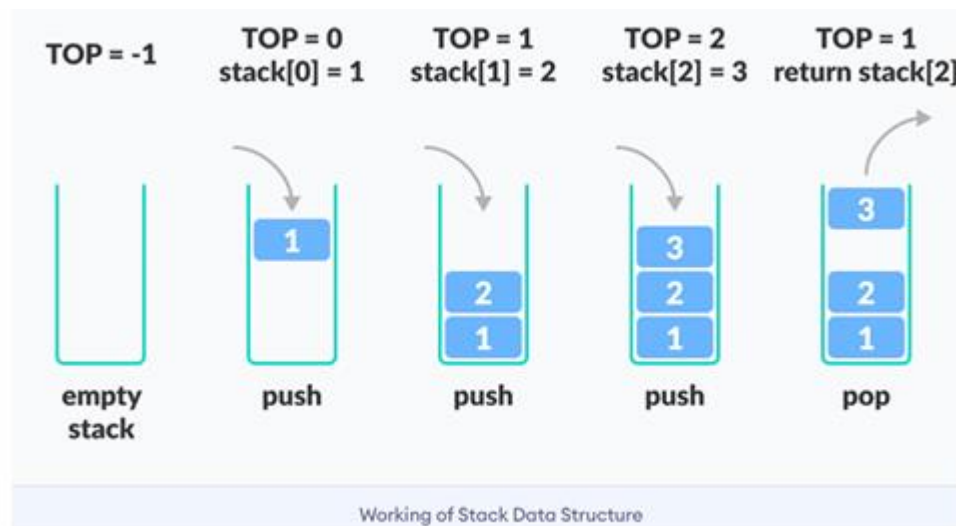
**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

A. TUJUAN PRAKTIKUM

- Mampu memahami konsep stack pada struktur data dan algoritma
- Mampu mengimplementasikan operasi-operasi pada stack
- Mampu memecahkan permasalahan dengan solusi stack

B. DASAR TEORI

Stack adalah struktur data sederhana yang digunakan untuk menyimpan data (mirip dengan Linked Lists). Dalam tumpukan, urutan kedatangan data penting. Sebuah tumpukan piring di kafetaria adalah contoh bagus dari tumpukan. Piring ditambahkan ke tumpukan saat mereka dibersihkan dan ditempatkan di bagian atas. Ketika sebuah piring dibutuhkan, diambil dari bagian atas tumpukan. Piring pertama yang ditempatkan di tumpukan adalah yang terakhir digunakan. Definisi: Sebuah tumpukan adalah daftar terurut di mana penyisipan dan penghapusan dilakukan di satu ujung, disebut atas. Elemen terakhir yang dimasukkan adalah yang pertama dihapus. Oleh karena itu, disebut daftar Last in First out (LIFO). Operasi pada stack melibatkan beberapa fungsi dasar yang dapat dilakukan pada struktur data ini. Berikut adalah beberapa operasi umum pada stack:



- **Push (Masukkan):** Menambahkan elemen ke dalam tumpukan pada posisi paling atas atau ujung.
- **Pop (Keluarkan):** Menghapus elemen dari posisi paling atas atau ujung tumpukan.
- **Top (Atas):** Mendapatkan nilai atau melihat elemen teratas pada tumpukan tanpa menghapusnya.
- **IsEmpty (Kosong):** Memeriksa apakah tumpukan kosong atau tidak.
- **IsFull (Penuh):** Memeriksa apakah tumpukan penuh atau tidak (terutama pada implementasi tumpukan dengan kapasitas terbatas).
- **Size (Ukuran):** Mengembalikan jumlah elemen yang ada dalam tumpukan.

- Peek (Lihat): Melihat nilai atau elemen pada posisi tertentu dalam tumpukan tanpa menghapusnya.
- Clear (Hapus Semua): Mengosongkan atau menghapus semua elemen dari tumpukan. i. Search (Cari): Mencari keberadaan elemen tertentu dalam tumpukan.

Guided (berisi screenshot source code & output program disertai penjelasannya)

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
PS C:\Users\USER\GitHub\Struktur-Data-6\Modul 7\Guided>

```

Deskripsi:

Pada guided 1, program ini adalah implementasi struktur data stack menggunakan array. Program ini memiliki beberapa fungsi utama seperti pushArrayBuku (menambahkan data ke stack), popArrayBuku (menghapus data dari stack), peekArrayBuku (melihat data

pada posisi tertentu di stack), countStack (menghitung jumlah data di stack), changeArrayBuku (mengubah data pada posisi tertentu di stack), destroyArraybuku (menghapus semua data di stack), dan cetakArrayBuku (mencetak semua data di stack).

E. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

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 << "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;
} else {
    cout << "Kalimat tersebut bukan palindrom" << endl;
}

return 0;
}
}

```

Screenshots Output

```

Masukkan Kalimat: mamam
Kalimat tersebut adalah palindrom
PS C:\Users\USER\GitHub\Struktur-Data-6\Modul 7\Unguided>

```

```

Masukkan Kalimat: fadhel yussie ramadhan
Kalimat tersebut bukan palindrom
PS C:\Users\USER\GitHub\Struktur-Data-6\Modul 7\Unguided>

```

Deskripsi:

Pada unguided 1, program ini digunakan untuk memeriksa apakah sebuah kalimat merupakan palindrom atau bukan. Program meminta pengguna untuk memasukkan sebuah kalimat, kemudian menyimpan setiap karakter dari kalimat tersebut ke dalam stack. Setelah itu, program akan membandingkan setiap karakter dari kalimat dengan karakter yang diambil dari stack secara terbalik. Jika setiap karakter sama, maka kalimat tersebut adalah palindrom, jika tidak, maka kalimat bukan palindrom.

F. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

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

```

Masukkan Kalimat: 2311102322
Hasil: 2232011132
PS C:\Users\USER\GitHub\Struktur-Data-6\Modul 7\Unguided>

```

Deskripsi:

Pada unguided 2, program ini digunakan untuk membalik urutan sebuah kalimat yang dimasukkan oleh pengguna. Program meminta pengguna untuk memasukkan sebuah kalimat, kemudian menyimpan setiap karakter dari kalimat tersebut ke dalam stack. Setelah itu, program akan mengambil setiap karakter dari stack dan mencetaknya sehingga urutan kalimat menjadi terbalik.

G. Kesimpulan

Stack adalah struktur data linear yang bersifat Last In First Out (LIFO). Operasi utama pada stack meliputi push (menambahkan data ke stack), pop (menghapus data dari stack), dan peek (melihat data teratas di stack tanpa menghapusnya). Stack dapat diimplementasikan dengan menggunakan array atau linked list. Dalam praktikum ini, kita telah mempelajari implementasi stack menggunakan array, serta beberapa aplikasi sederhana dari stack seperti memeriksa palindrom, dan membalik urutan kalimat. Pemahaman konsep stack sangat penting dalam algoritma dan pemrograman, karena banyak algoritma yang memanfaatkan struktur data ini, seperti algoritma pencarian, pengurutan, dan lain-lain.

Referensi

Modul 6 Stack praktikum algoritma struktur data IF 11 B whatsapp grup Praktikum strukdat IF-11-B.