

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL 8
PENGENALAN CODE BLOCKS**



Disusun Oleh :

NAMA : Muhamad Ilham Syahid
NIM : 103112400155

Dosen
WAHYU ANDI SAPUTRA

**PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

Stack adalah struktur data linear yang menggunakan prinsip LIFO (Last In First Out), di mana elemen terakhir yang dimasukkan akan menjadi elemen pertama yang dikeluarkan. Operasi utama pada stack adalah push untuk menambahkan data dan pop untuk mengambil data, yang keduanya hanya dapat dilakukan pada bagian TOP. Stack dapat diimplementasikan menggunakan pointer maupun array, dengan perbedaan pada pengelolaan memori dan batas jumlah data.

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

Guided 1

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

#define MAX 20

typedef int infotype;

struct Stack {
    infotype info[MAX];
    int top;
};

// Create stack
void createStack(Stack &S) {
    S.top = -1;
}

// Check empty
bool isEmpty(Stack S) {
    return S.top == -1;
}

// Push
void push(Stack &S, infotype x) {
    if (S.top < MAX - 1) {
        S.top++;
        S.info[S.top] = x;
    }
}

// Pop
infotype pop(Stack &S) {
    if (!isEmpty(S)) {
        infotype x = S.info[S.top];
        S.top--;
        return x;
    }
}
```

```
        return -1;
    }

// Print stack
void printInfo(Stack S) {
    if (isEmpty(S)) {
        cout << "Stack kosong" << endl;
    } else {
        for (int i = S.top; i >= 0; i--) {
            cout << S.info[i] << " ";
        }
        cout << endl;
    }
}

// Reverse stack
void balikStack(Stack &S) {
    Stack temp;
    createStack(temp);
    while (!isEmpty(S)) {
        push(temp, pop(S));
    }
    S = temp;
}

// Push ascending
void pushAscending(Stack &S, infotype x) {
    Stack temp;
    createStack(temp);

    while (!isEmpty(S) && S.info[S.top] > x) {
        push(temp, pop(S));
    }
    push(S, x);
    while (!isEmpty(temp)) {
        push(S, pop(temp));
    }
}

// Input stream
void getInputStream(Stack &S) {
    char ch;
    cout << "Masukkan karakter (Enter untuk selesai): ";
    while ((ch = cin.get()) != '\n') {
        push(S, ch);
    }
}
```

```
int main() {
    Stack S;
    createStack(S);

    // Latihan 1
    push(S, 3);
    push(S, 4);
    push(S, 8);
    pop(S);
    push(S, 2);
    push(S, 3);
    pop(S);
    push(S, 9);
    printInfo(S);

    cout << "balik stack" << endl;
    balikStack(S);
    printInfo(S);

    return 0;
}
```

Screenshots Output

```
PS C:\Pratikum Struktur Data\modul 7 laprak> cd "c:\Pratikum Struktur Data\modul 7 laprak\" ; if ($?) { g++ stack.cpp -o stack } ; if ($?) { .\stack }
9 2 4 3
balik stack
3 4 2 9
PS C:\Pratikum Struktur Data\modul 7 laprak>
```

Deskripsi:

Program ini merupakan implementasi ADT Stack menggunakan array dengan prinsip LIFO. Stack memiliki variabel top sebagai penunjuk elemen teratas. Operasi utama yang digunakan adalah push untuk menambah data dan pop untuk mengambil data dari stack. Program juga menyediakan fungsi untuk menampilkan isi stack (printInfo), membalik urutan stack (balikStack), memasukkan data secara terurut (pushAscending), serta menerima input hingga tombol Enter ditekan. Seluruh fungsi diuji pada main sesuai latihan Modul 7.

Kesimpulan

Stack merupakan struktur data linear yang bekerja dengan prinsip LIFO (Last In First Out), di mana operasi hanya dapat dilakukan pada elemen teratas (TOP). Melalui praktikum Modul 7, stack dapat diimplementasikan menggunakan array dengan operasi utama push dan pop. Implementasi ini membantu memahami cara penyimpanan, pengambilan, dan pengolahan data secara terstruktur serta melatih pemahaman konsep dasar struktur data dalam pemrograman.

C. Referensi

Modul Struktur Data – Modul 8: Queue