

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

**MODUL 7  
STACK**



**Disusun Oleh :**

NAMA : Muhammad Luthfi Arrafi

Ramadhani

NIM : 103112430043

**Dosen Pengampu:**

Fahrudin Mukti Wibowo

**PROGRAM STUDI S1 TEKNIK INFORMATIKA**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2025**

## A. Dasar Teori

Dasar teori pada modul ini membahas konsep Stack, yaitu struktur data data linier yang menerapkan prinsip LIFO (Last In First Out), di mana elemen terakhir yang masuk akan diambil pertama. Stack memiliki penunjuk Top untuk menunjukkan elemen paling atas, dengan dua operasi utama yaitu push (menambah data) dan pop (menghapus data). Stack dapat diimplementasikan menggunakan array atau pointer, serta dilengkapi fungsi tambahan seperti createStack() untuk membuat stack baru, isEmpty() untuk memeriksa apakah stack kosong, dan balikStack() untuk membalik urutan elemen di dalam stack. Struktur ini sering digunakan di berbagai proses seperti pemanggilan fungsi, fitur undo-redo, maupun pengolahan data di dalam program.

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

### Guided 1

(stack.cpp)

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

struct Node
{
    int data;
    Node *next;
};

bool isEmpty(Node *top)
{
    return top == nullptr;
}

void push(Node *&top, int data)
{
    Node *newNode = new Node();
    newNode->data = data;
    newNode->next = top;
    top = newNode;
}

int pop(Node *&top)
{
    if (isEmpty(top))
    {
        cout << "Stack kosong, tidak dapat pop!" << endl;
        return 0;
    }

    int poppedData = top->data;
```

```
Node *temp = top;
top = top->next;

delete temp;
return poppedData;
}

void show(Node *top)
{
    if (isEmpty(top))
    {
        cout << "Stack Kosong." << endl;
        return;
    }

    cout << "TOP -> ";
    Node *temp = top;

    while (temp != nullptr)
    {
        cout << temp->data << " -> ";
        temp = temp->next;
    }

    cout << "NULL" << endl;
}

int main()
{
    Node *stack = nullptr;

    push(stack, 10);
    push(stack, 20);
    push(stack, 30);

    cout << "Menampilkan isi stack:" << endl;
    show(stack);

    cout << "Pop: " << pop(stack) << endl;

    cout << "Menampilkan sisa stack:" << endl;
    show(stack);

    return 0;
}
```

## Screenshots Output

```
PS E:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043> cd "e:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043\Minggu 6\Guided\" ; if ($?) { g++ stack.cpp -o stack } ; if ($?) { .\stack }
Menampilkan isi stack:
TOP -> 30 -> 20 -> 10 -> NULL
Pop: 30
Menampilkan sisa stack:
TOP -> 20 -> 10 -> NULL
PS E:\BACKUP092024\VSCODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043\Minggu 6\Guided>
```

Deskripsi:

Program tersebut merupakan contoh penerapan stack dengan linked list sederhana. Setiap Node menyimpan data serta penunjuk ke elemen di bawahnya. Fungsi *push* menambah elemen baru di bagian atas dengan memperbarui pointer *top*, sedangkan *pop* mengambil dan menghapus elemen teratas sekaligus mengosongkan memorinya. Fungsi *isEmpty* digunakan untuk memeriksa apakah stack kosong, dan *show* menampilkan seluruh isi stack dari elemen paling atas hingga akhir. Pada bagian *main*, program menambahkan data 10, 20, dan 30 ke dalam stack, lalu melakukan satu operasi *pop*, kemudian menampilkan kondisi stack sebelum dan sesudah proses tersebut.

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

Unguided 1

(stack.h)

```
// Muhammad Luthfi Arrafi Ramadhani
// 103112430043
// IF 12-06

#ifndef STACK_H
#define STACK_H

const int MAX = 20;
typedef int infotype;

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

void CreateStack(Stack &S);
void Push(Stack &S, infotype x);
infotype Pop(Stack &S);
void printInfo(const Stack &S);
void balikStack(Stack &S);

#endif
```

Deskripsi:

File stack.h ini berisi deklarasi tipe data dan fungsi yang digunakan untuk membuat struktur stack berbasis array. Di dalamnya terdapat konstanta MAX sebagai batas ukuran stack, serta struktur *Stack* yang menyimpan data bertipe infotype dan variabel *top* untuk menandai posisi elemen teratas. Selain itu, file ini juga mendefinisikan beberapa fungsi dasar seperti *CreateStack()*, *Push()*, *Pop()*, *printInfo()*, dan *balikStack()* yang akan diimplementasikan pada file lain.

(stack.cpp)

```
// Muhammad Luthfi Arrafi Ramadhani
// 103112430043
// IF 12-06

#include "stack.h"
#include <iostream>
using namespace std;

void CreateStack(Stack &S) {
    S.top = -1;
}

void Push(Stack &S, infotype x) {
    if (S.top >= MAX - 1) {
        cout << "Stack penuh, tidak dapat push!" << endl;
        return;
    }
    S.top++;
    S.info[S.top] = x;
}

infotype Pop(Stack &S) {
    if (S.top == -1) {
        cout << "Stack kosong, tidak dapat pop!" << endl;
        return 0;
    }
    infotype val = S.info[S.top];
    S.top--;
    return val;
}

void printInfo(const Stack &S) {
    if (S.top == -1) {
        cout << "[TOP]" << endl;
        return;
    }
    cout << "[TOP] ";
    for (int i = S.top; i >= 0; --i) {
```

```

        cout << S.info[i];
        if (i > 0) cout << " ";
    }
    cout << endl;
}

void balikStack(Stack &S) {
    int i = 0, j = S.top;
    while (i < j) {
        infotype tmp = S.info[i];
        S.info[i] = S.info[j];
        S.info[j] = tmp;
        i++; j--;
    }
}

```

Deskripsi:

File stack.cpp ini berisi implementasi dari fungsi-fungsi yang telah dideklarasikan di stack.h. Fungsi *CreateStack()* menginisialisasi stack agar kosong, *Push()* menambah elemen baru ke bagian atas selama stack belum penuh, dan *Pop()* menghapus serta mengembalikan elemen paling atas. Fungsi *printInfo()* menampilkan isi stack dari atas ke bawah, sedangkan *balikStack()* membalik urutan data dengan menukar posisi antar elemen.

(main.cpp)

```

// Muhammad Luthfi Arrafi Ramadhani
// 103112430043
// IF 12-06

#include <iostream>
#include "stack.h"
using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    Stack S;
    CreateStack(S);
    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 E:\BACKUP092024\VS CODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043\Minggu 6\Unguided> g++ main.cpp stack.cpp -o main
PS E:\BACKUP092024\VS CODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043\Minggu 6\Unguided> ./main
Hello world!
[TOP] 9 2 4 3
balik stack
[TOP] 3 4 2 9
PS E:\BACKUP092024\VS CODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhani-103112430043\Minggu 6\Unguided> █

```

## Deskripsi:

File main.cpp ini berfungsi untuk menjalankan program utama operasi stack. Program dimulai dengan membuat stack kosong, kemudian melakukan operasi *Push()* dan *Pop()*. Setelah menampilkan isi stack menggunakan *printInfo()*, fungsi *balikStack()* dijalankan untuk membalik urutan elemen, dan hasil akhirnya ditampilkan sesuai dengan output yang diinginkan.

## Unguided 2

(stack.h)

```

// Muhammad Luthfi Arrafi Ramadhani
// 103112430043
// IF 12-06

#ifndef STACK_H
#define STACK_H

const int MAX = 20;
typedef int infotype;

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

void CreateStack(Stack &S);
void Push(Stack &S, infotype x);
infotype Pop(Stack &S);
void printInfo(const Stack &S);
void balikStack(Stack &S);
void pushAscending(Stack &S, infotype x);

#endif

```

(stack.cpp)

```
// Muhammad Luthfi Arrafi Ramadhani
// 103112430043
// IF 12-06

#include "stack.h"
#include <iostream>
using namespace std;

void CreateStack(Stack &S) {
    S.top = -1;
}

void Push(Stack &S, infotype x) {
    if (S.top >= MAX - 1) {
        cout << "Stack penuh, tidak dapat push!" << endl;
        return;
    }
    S.top++;
    S.info[S.top] = x;
}

void pushAscending(Stack &S, infotype x) {
    if (S.top >= MAX - 1) {
        cout << "Stack penuh, tidak dapat push!" << endl;
        return;
    }
    int j = S.top;

    while (j >= 0 && S.info[j] > x) {
        S.info[j + 1] = S.info[j];
        j--;
    }
    S.info[j + 1] = x;
    S.top++;
}

infotype Pop(Stack &S) {
    if (S.top == -1) {
        cout << "Stack kosong, tidak dapat pop!" << endl;
        return 0;
    }
    infotype val = S.info[S.top];
    S.top--;
    return val;
}

void printInfo(const Stack &S) {
```

```

if (S.top == -1) {
    cout << "[TOP]" << endl;
    return;
}
cout << "[TOP] ";
for (int i = S.top; i >= 0; --i) {
    cout << S.info[i];
    if (i > 0) cout << " ";
}
cout << endl;
}

void balikStack(Stack &S) {
    int i = 0, j = S.top;
    while (i < j) {
        infotype tmp = S.info[i];
        S.info[i] = S.info[j];
        S.info[j] = tmp;
        i++; j--;
    }
}

```

(main.cpp)

```

// Muhammad Luthfi Arrafi Ramadhani
// 103112430043
// IF 12-06

#include <iostream>
#include "stack.h"
using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    Stack S;
    CreateStack(S);
    pushAscending(S,3);
    pushAscending(S,4);
    pushAscending(S,8);
    pushAscending(S,2);
    pushAscending(S,3);
    pushAscending(S,9);
    printInfo(S);
    cout << "balik stack" << endl;
    balikStack(S);
    printInfo(S);
}

```

```
    return 0;  
}
```

## Screenshots Output

```
PS E:\BACKUP092024\VS CODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhan-103112430043\Minggu 6\Unguided> g++ main.cpp stack.cpp -o main  
PS E:\BACKUP092024\VS CODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhan-103112430043\Minggu 6\Unguided> ./main  
Hello world!  
[TOP] 9 8 4 3 3 2  
balik stack  
[TOP] 2 3 3 4 8 9  
PS E:\BACKUP092024\VS CODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhan-103112430043\Minggu 6\Unguided>
```

## Deskripsi:

Program ini menambahkan prosedur pushAscending() untuk memasukkan data ke dalam stack secara berurutan dari nilai terkecil hingga terbesar. Saat data baru dimasukkan, program akan menyesuaikan posisi agar tetap berada pada urutan yang sesuai. Setelah semua data ditambahkan, program menampilkan isi stack dan menggunakan fungsi balikStack() untuk membalik urutan elemen sebelum menampilkannya kembali.

## Unguided 3

(stack.h)

```
// Muhammad Luthfi Arrafi Ramadhan  
// 103112430043  
// IF 12-06  
  
#ifndef STACK_H  
#define STACK_H  
  
const int MAX = 20;  
typedef int infotype;  
  
struct Stack {  
    infotype info[MAX];  
    int top;  
};  
  
void CreateStack(Stack &S);  
void Push(Stack &S, infotype x);  
infotype Pop(Stack &S);  
void printInfo(const Stack &S);  
void balikStack(Stack &S);  
void pushAscending(Stack &S, infotype x);  
void getInputStream(Stack &S);  
  
#endif
```

(stack.cpp)

```
// Muhammad Luthfi Arrafi Ramadhani
// 103112430043
// IF 12-06

#include "stack.h"
#include <iostream>
#include <cctype>
using namespace std;

void CreateStack(Stack &S) {
    S.top = -1;
}

void Push(Stack &S, infotype x) {
    if (S.top >= MAX - 1) {
        cout << "Stack penuh, tidak dapat push!" << endl;
        return;
    }
    S.top++;
    S.info[S.top] = x;
}

void pushAscending(Stack &S, infotype x) {
    if (S.top >= MAX - 1) {
        cout << "Stack penuh, tidak dapat push!" << endl;
        return;
    }
    int j = S.top;

    while (j >= 0 && S.info[j] > x) {
        S.info[j + 1] = S.info[j];
        j--;
    }
    S.info[j + 1] = x;
    S.top++;
}

infotype Pop(Stack &S) {
    if (S.top == -1) {
        cout << "Stack kosong, tidak dapat pop!" << endl;
        return 0;
    }
    infotype val = S.info[S.top];
    S.top--;
    return val;
}

void printInfo(const Stack &S) {
    if (S.top == -1) {
```

```

        cout << "[TOP]" << endl;
        return;
    }
    cout << "[TOP] ";
    for (int i = S.top; i >= 0; --i) {
        cout << S.info[i];
        if (i > 0) cout << " ";
    }
    cout << endl;
}

void balikStack(Stack &S) {
    int i = 0, j = S.top;
    while (i < j) {
        infotype tmp = S.info[i];
        S.info[i] = S.info[j];
        S.info[j] = tmp;
        i++; j--;
    }
}

void getInputStream(Stack &S) {
    char ch;
    while (true) {
        ch = cin.get();
        if (ch == '\n' || ch == EOF) break;
        if (isdigit(static_cast<unsigned char>(ch))) {
            Push(S, static_cast<infotype>(ch - '0'));
        }
    }
}

```

(main.cpp)

```

// Muhammad Luthfi Arrafi Ramadhan
// 103112430043
// IF 12-06

#include <iostream>
#include "stack.h"
using namespace std;

int main()
{
    cout << "Hello world!" << endl;
    Stack S;
    CreateStack(S);

```

```
getInputStream(S);
printInfo(S);
cout<<"balik stack"<<endl;
balikStack(S);
printInfo(S);
return 0;
}
```

## Screenshots Output

```
PS E:\BACKUP092024\VS CODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhan-103112430043\Minggu 6\Unguided> g++ main.cpp stack.cpp -o main
PS E:\BACKUP092024\VS CODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhan-103112430043\Minggu 6\Unguided> ./main
Hello world!
4729601
[TOP] 1 0 6 9 2 7 4
balik stack
[TOP] 4 7 2 9 6 0 1
PS E:\BACKUP092024\VS CODE\3rd sm\Praktikum Muhammad Luthfi Arrafi Ramadhan-103112430043\Minggu 6\Unguided>
```

### Deskripsi:

Program ini menambahkan prosedur getInputStream() untuk membaca input dari pengguna secara langsung dan menyimpannya ke dalam stack sampai pengguna menekan tombol Enter. Setiap karakter yang dimasukkan akan diubah menjadi angka lalu dimasukkan satu per satu menggunakan fungsi Push(). Dan hasilnya akan ditampilkan sesuai input dan urutannya kemudian dibalik menggunakan fungsi balikStack().

## D. Kesimpulan

Dari ketiga latihan pada materi Stack ini dapat disimpulkan bahwa struktur data stack memudahkan pengelolaan data dengan konsep LIFO (Last In First Out), di mana data terakhir yang dimasukkan akan diambil terlebih dahulu. Melalui penerapan fungsi Push, Pop, pushAscending(), dan getInputStream(), dapat dipahami cara menambahkan, menghapus, serta mengatur urutan data di dalam stack. Selain itu, penggunaan balikStack() menunjukkan bagaimana urutan elemen dalam stack dapat dibalik.

## E. Referensi

Setiyawan, R. D., Hermawan, D., Abdillah, A. F., Mujayanah, A., & Vindua, R. (2024). Penggunaan struktur data stack dalam pemrograman C++ dengan pendekatan array dan linked list. *Journal of Education and Technology*, 5(2), 484–498. <https://doi.org/10.31932/jutech.v5i2.4263>

Amaylia, S., Setiabud, V. A., Alvianino, R., Saputra, R. N., Wardhani, H. K., & Suroni, A. (2025). Application of stack data structure in application development. *Journal of Advanced Systems Intelligence and Cybersecurity*, 1(1). <https://journal.unesa.ac.id/index.php/jasic/article/view/44459>