

LAPORAN PRAKTIKUM

MODUL VI STACK



Disusun oleh:
Muhammad Rifki Fadhilah
NIM: 2311102032

Dosen Pengampu:
Wahyu Andi Saputra, S.Pd., M.Eng.

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

BAB I

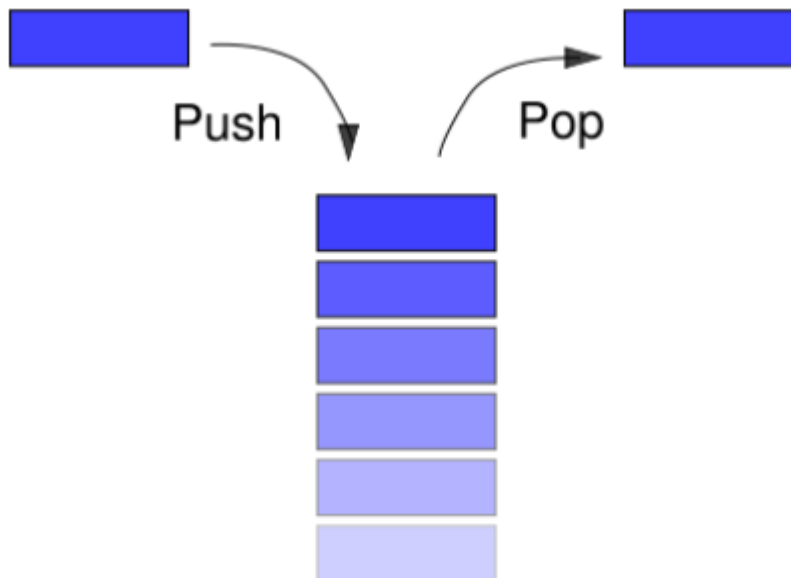
TUJUAN PRAKTIKUM

1. Mahasiswa mampu menjelaskan definisi dan konsep dari Hash Code
2. Mahasiswa mampu menerapkan Hash Code kedalam pemrograman

BAB II

DASAR TEORI

Stack adalah sebuah kumpulan data dimana data yang diletakkan di atas data yang lain. Dengan demikian stack adalah struktur data yang menggunakan konsep LIFO (Last In First Out). Sebagai contoh, kamu mempunyai empat buku, yaitu buku matematika, fisika, biologi, dan kimia. Buku matematika kamu masukkan terlebih dahulu, lalu buku selanjutnya fisika, biologi, dan kimia. Nah, untuk mendapatkan buku fisika, kamu harus mengeluarkan buku kimia dan biologi terlebih dahulu, karena kedua buku tersebut tergolong buku yang terakhir masuk. Analogi tersebut seperti halnya stack yang memiliki sifat LIFO. Dengan demikian, elemen terakhir yang disimpan dalam stack menjadi elemen pertama yang diambil. Dalam proses komputasi, untuk meletakkan sebuah elemen pada bagian atas dari stack, maka dilakukan operasi push. Dan untuk memindahkan dari tempat yang atas tersebut, maka dilakukan operasi pop.



Kelebihan:

1. Membantu mengelola data dengan metode LIFO
2. Secara otomatis membersihkan objek
3. Tidak mudah rusak
4. Ukuran variabel tidak dapat diubah
5. Mengontrol memori secara mandiri

Kekurangan:

1. Memori stack sangat terbatas
2. Ada kemungkinan stack akan meluap atau overflow jika terlalu banyak objek
3. Tidak memungkinkan akses acak, karena harus mengeluarkan tumpukan paling atas terlebih dahulu untuk mengakses tumpukan paling bawah

Jenis-Jenis Operasi Stack:

1. Pop
Operasi Pop pada stack adalah operasi yang berfokus pada penghapusan elemen.
2. Push
Operasi Push berfokus pada memasukkan elemen ke dalam stack atau tumpukan.
3. isFull
Operasi stack yang satu ini adalah untuk mengetahui apakah tumpukan sudah penuh atau belum.
4. isEmpty
isEmpty merupakan operasi yang digunakan untuk memeriksa apakah tumpukan kosong atau tidak.
5. Peek
Operasi Peek atau mengintip adalah operasi yang dilakukan untuk mengetahui data teratas dari tumpukan tanpa harus menghapusnya.

Stack dapat diimplementasikan dengan menggunakan array atau linked list. Kelebihan dari implementasi stack menggunakan array adalah mudah untuk diimplementasikan. Sementara itu, kekurangannya adalah bisa dibilang tidak terlalu dinamis. Pasalnya, hal itu tidak tumbuh dan menyusut, tergantung pada kebutuhan saat proses. Jika linked list, kelebihanannya adalah dapat bertambah dan berkurang sesuai dengan kebutuhan saat runtime. Sementara kekurangannya adalah harus membutuhkan memori ekstra karena ada pointer di dalamnya.

BAB III

GUIDED

1. 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");
pushArrayBuku("Indonesia");
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;
}

```

Screenshoot program

```

PS D:\Project VS Code\C++\Semester 2\Praktikum Struktur Data\modul6\output> cd 'd:\Project VS Code\C++\Semester 2\Praktikum
struktur Data\modul6\output'
PS D:\Project VS Code\C++\Semester 2\Praktikum Struktur Data\modul6\output> & .\'guided1.exe'
Data telah penuh
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 D:\Project VS Code\C++\Semester 2\Praktikum Struktur Data\modul6\output>

```


Deskripsi program

Program diatas adalah implementasi dari Stack menggunakan array. Dalam program ini, terdapat arrayBuku yang menampung 5 elemen. Terdapat juga variabel maksimal yang bernilai 5 untuk kapasitas maksimum array dan top bernilai 0 untuk menunjukkan posisi teratas dari stack. .Selanjutnya ada fungsi-fungsi utama dari program ini yaitu, isFull untuk memeriksa apakah stack penuh, isEmpty untuk memeriksa apakah stack kosong, pushArrayBuku untuk menambah elemen ke stack, popArrayBuku untuk menghapus elemen teratas dari stack, peekArrayBuku untuk melihat elemen pada posisi tertentu dari atas stack, changeArrayBuku untuk mengubah elemen pada posisi tertentu dalam stack, destroyArraybuku untuk menghapus semua elemen dalam stack, cetakArrayBuku untuk menampilkan semua elemen dalam stack, countStack untuk menghitung jumlah dalam stack. Di fungsi main, terdapat beberapa operasi seperti menambahkan data ke dalam stack menggunakan pushArrayBuku(), mencetak isi stack menggunakan cetakArrayBuku(), mengubah data pada posisi tertentu menggunakan changeArrayBuku(), menghapus data dari stack menggunakan popArrayBuku(), dan lain-lain. Dan akhirnya, penghapusan seluruh data dalam stack menggunakan destroyArraybuku()

LATIHAN KELAS - UNGUIDED

1. Unguided 1

Source code

```
#include <iostream>
using namespace std;
char charArray[100];
int maksimal = 100, top = 0;
bool isFull()
{
    return (top == maksimal);
}
bool isEmpty()
{
    return (top == 0);
}
void push(char data)
{
    if (isFull())
    {
        cout << "Stack telah penuh" << endl;
    }
    else
    {
        charArray[top] = data;
        top++;
    }
}
char pop()
{
    if (isEmpty())
    {
        cout << "Stack kosong" << endl;
```

```

        return -1; // Mengembalikan karakter khusus untuk
menandai kesalahan
    }
    else
    {
        top--;
        char popped = charArray[top]; // Mengambil karakter
sebelum mengurangi top
        return popped;
    }
}

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

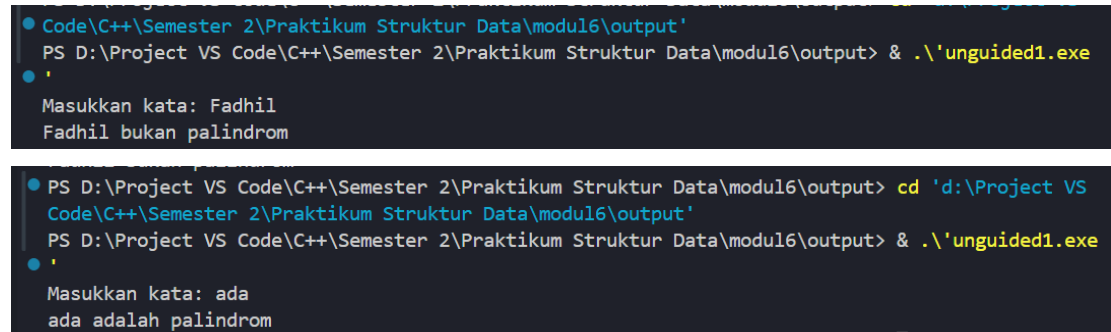
    for (int i = 0; i < kata.length(); i++) {
        push(kata[i]);
    }

    string kataDibalik = "";
    for (int i = 0; i < kata.length(); i++)
    {
        kataDibalik.push_back(pop());
    }

    if (kata == kataDibalik){
        cout << kata << " adalah palindrom" << endl;
    } else {
        cout << kata << " bukan palindrom" << endl;
    }
    return 0;
}

```

Screenshoot program



```
Code\C++\Semester 2\Praktikum Struktur Data\modul6\output'
PS D:\Project VS Code\C++\Semester 2\Praktikum Struktur Data\modul6\output> & .\'unguided1.exe
'
Masukkan kata: Fadhil
Fadhil bukan palindrom

PS D:\Project VS Code\C++\Semester 2\Praktikum Struktur Data\modul6\output> cd 'd:\Project VS
Code\C++\Semester 2\Praktikum Struktur Data\modul6\output'
PS D:\Project VS Code\C++\Semester 2\Praktikum Struktur Data\modul6\output> & .\'unguided1.exe
'
Masukkan kata: ada
ada adalah palindrom
```

Deskripsi program

Program ini adalah contoh implementasi dari konsep stack untuk mengecek apakah sebuah kata merupakan palindrom atau bukan. Di program ini, kita menggunakan array `charArray` untuk menyimpan karakter-karakter dari kata yang dimasukkan. Variabel `top` digunakan untuk menunjukkan posisi paling atas dari stack, dan maksimal adalah ukuran maksimum dari stack. Seperti di `guided1` kita menggunakan fungsi utama stack yaitu fungsi `isfull`, `isEmpty`, `push`, dan `pop`. Di dalam fungsi `main`, program akan meminta pengguna memasukkan sebuah kata, dan setiap karakter dari kata tersebut akan dimasukkan ke dalam stack menggunakan fungsi `push`. Selanjutnya, program akan mengambil karakter-karakter dari stack dan membangun kata yang dibalik menggunakan loop `for`. Kata yang dibalik tersebut akan dibandingkan dengan kata aslinya. Jika kedua kata sama, maka kata tersebut adalah palindrom. Jika tidak, maka bukan palindrom.

2. Unguided 2

Source code

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

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

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

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

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

char pop()
{
    if (isEmpty())
```

```

    {
        cout << "Stack kosong" << endl;
        return -1; // Mengembalikan karakter khusus untuk
menandai kesalahan
    }
    else
    {
        top--;
        char popped = charArray[top]; // Mengambil karakter
sebelum mengurangi top
        charArray[top] = '\0'; // Menghapus karakter dari stack
        return popped;
    }
}

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

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

    for (int i = 0; i < input.length(); i++)

```

```

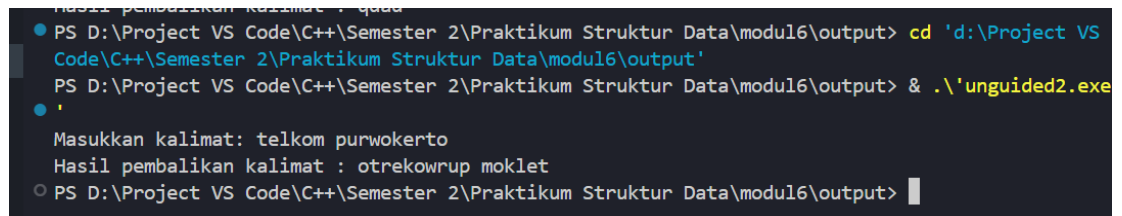
    {
        push(input[i]);
    }

    cout << "Hasil pembalikan kalimat : ";
    while (!isEmpty())
    {
        cout << peek();
        pop();
    }

    return 0;
}

```

Screenshoot program



```

PS D:\Project VS Code\C++\Semester 2\Praktikum Struktur Data\modul6\output> cd 'd:\Project VS Code\C++\Semester 2\Praktikum Struktur Data\modul6\output'
PS D:\Project VS Code\C++\Semester 2\Praktikum Struktur Data\modul6\output> & .\'unguided2.exe
Masukkan kalimat: telkom purwokerto
Hasil pembalikan kalimat : otrekowrup moklet
PS D:\Project VS Code\C++\Semester 2\Praktikum Struktur Data\modul6\output>

```

Deskripsi program

Program di atas merupakan contoh implementasi stack dalam C++ untuk membalikkan sebuah kalimat. Di program tersebut, kita menggunakan array `charArray` untuk menyimpan karakter-karakter dari kalimat yang akan dibalik. Variabel `top` digunakan untuk menandai indeks teratas (atau puncak) dari stack, sedangkan maksimal adalah kapasitas maksimum dari stack. Sama seperti `guided 1` dan `unguided 1` kita menggunakan beberapa fungsi utama stack yaitu `isFull`, `isEmpty`, `push`, `pop`, `peek`. Di dalam fungsi `main`, program akan meminta input kalimat dari pengguna. Setiap karakter dari kalimat tersebut akan dimasukkan ke dalam stack menggunakan fungsi `push()`. Setelah itu, program akan menampilkan

hasil pembalikan kalimat dengan mengeluarkan karakter teratas dari stack menggunakan fungsi peek() dan pop() sampai stack kosong.

BAB IV

KESIMPULAN

Stack adalah struktur data yang mengikuti konsep Last In First Out (LIFO), di mana elemen terakhir yang dimasukkan adalah elemen pertama yang diambil. Operasi push digunakan untuk menambah elemen ke dalam stack, sedangkan operasi pop digunakan untuk menghapus elemen dari stack. Stack memiliki kelebihan dalam mengelola data dengan metode LIFO, membersihkan objek secara otomatis, dan mengontrol memori secara mandiri. Namun, stack juga memiliki kekurangan, seperti memori yang sangat terbatas, kemungkinan overflow jika terlalu banyak objek, dan tidak memungkinkan akses acak.

Jenis-jenis operasi stack meliputi pop (penghapusan elemen), push (penambahan elemen), isFull (cek apakah stack penuh), isEmpty (cek apakah stack kosong), dan peek (melihat elemen teratas tanpa menghapusnya). Stack dapat diimplementasikan menggunakan array atau linked list. Implementasi stack dengan array mudah, tetapi tidak dinamis. Implementasi dengan linked list lebih dinamis, tetapi membutuhkan memori ekstra karena adanya pointer.

BAB V

DAFTAR PUSTAKA

1. Asisten Praktikum. (2024). Modul VI : Stack
2. Adieb,Maulana. (2021). Memahami Stack, Struktur Data yang Mudahkan Programmer Jalankan Tugasnya. Diakses pada 21 Mei 2024, dari <https://glints.com/id/lowongan/stack-adalah/>
3. nblognlife. (26 April 2014). Stack pada C++. Diakses pada 21 Mei 2024, dari <https://www.nblognlife.com/2014/04/stack-pada-c.html>