

LAPORAN PRAKTIKUM

MODUL 6

STACK



Disusun oleh:

Annisa Al Jauhar

NIM: 2311102014

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

A. TUJUAN PRAKTIKUM

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

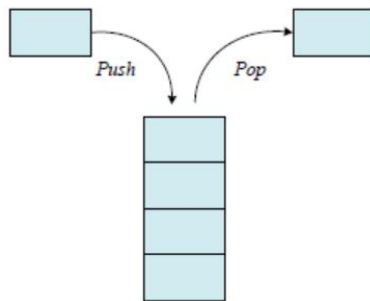
BAB II

DASAR TEORI

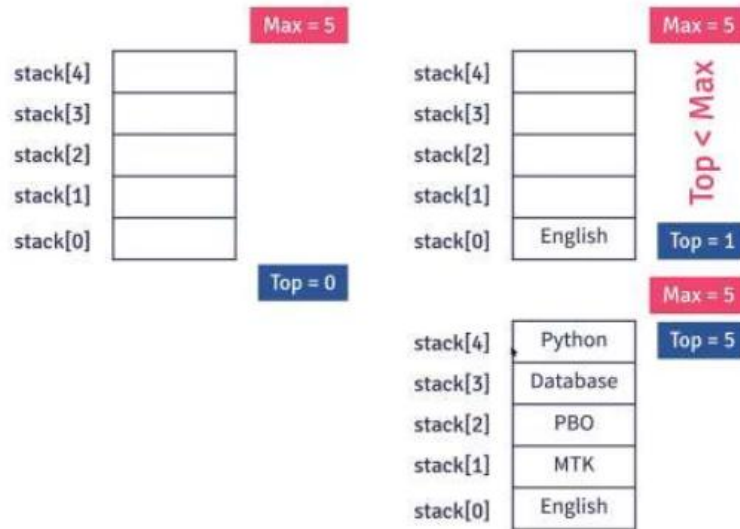
B. DASAR TEORI

a) Pengertian Stack

Stack merupakan sebuah kumpulan data yang diletakkan di atas data lainnya, seperti sebuah tumpukan. Dengan demikian, stack merupakan salah satu struktur data yang menerapkan prinsip LIFO (Last In First Out). Dimana elemen yang terakhir disimpan dalam stack, menjadi elemen yang pertama diambil. Untuk meletakkan sebuah elemen pada bagian atas dari stack, maka dilakukan operasi push. Sedangkan untuk memindahkan sebuah elemen dari tempat atas tersebut dalam sebuah stack, maka dilakukan operasi pop.



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.

e. IsFull (Penuh): Memeriksa apakah tumpukan penuh atau tidak (terutama pada implementasi tumpukan dengan kapasitas terbatas).

f. Size (Ukuran): Mengembalikan jumlah elemen yang ada dalam tumpukan.

g. Peek (Lihat): Melihat nilai atau elemen pada posisi tertentu dalam tumpukan tanpa menghapusnya.

h. Clear (Hapus Semua): Mengosongkan atau menghapus semua elemen dari tumpukan.

i. Search (Cari): Mencari keberadaan elemen tertentu dalam tumpukan.

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");
    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

```
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\annis\OneDrive\Desktop\Semester 2\C++ VsCode Praktikum\Struktur Data rabu\Modul 6>
```

Deskripsi program

Program tersebut merupakan implementasi stack menggunakan array. Stack digunakan untuk menyimpan judul buku dengan prinsip Last In, First Out (LIFO). Array `arrayBuku` digunakan untuk menyimpan data dengan batas maksimal 5, dan variabel `top` menunjukkan posisi teratas stack. Fungsi utilitas seperti `isFull()` dan `isEmpty()` memeriksa keadaan stack. Operasi dasar stack seperti `pushArrayBuku()` (menambah data), `popArrayBuku()` (menghapus data), `peekArrayBuku()` (melihat data pada posisi tertentu), dan `changeArrayBuku()` (mengubah data) diimplementasikan. Program diuji melalui fungsi `main()` dengan operasi-operasi pada stack. Ini menunjukkan penggunaan stack untuk menyimpan dan mengelola data, khususnya judul buku, secara efisien.

LATIHAN KELAS - UNGUIDED

1. Unguided 1

A. Source code

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

bool isPalindrome(string kalimat) {
    stack<char> stackKarakter;
    int length = kalimat.length();

    for (int i = 0; i < length / 2; i++) {
        stackKarakter.push(kalimat[i]);
    }

    int i = (length + 1) / 2;
    while (i < length) {
        if (kalimat[i] != stackKarakter.top()) {
            return false; // Tidak palindrom
        }
        stackKarakter.pop();
        i++;
    }

    return true; // Palindrom
}

int main() {
    string kalimat;

    cout << "Masukkan kalimat: ";
```

```

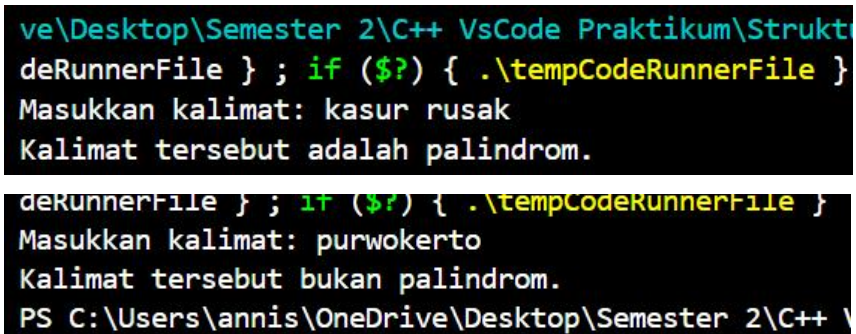
getline(cin, kalimat);

if (isPalindrome(kalimat)) {
    cout << "Kalimat tersebut adalah palindrom." << endl;
} else {
    cout << "Kalimat tersebut bukan palindrom." << endl;
}

return 0;
}

```

Screenshoot program



```

ve\Desktop\Semester 2\C++ VsCode Praktikum\Struktur
deRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Masukkan kalimat: kasur rusak
Kalimat tersebut adalah palindrom.

deRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Masukkan kalimat: purwokerto
Kalimat tersebut bukan palindrom.
PS C:\Users\annis\OneDrive\Desktop\Semester 2\C++ V

```

Deskripsi program

Program ini merupakan sebuah program yang bertujuan untuk mengecek apakah sebuah kalimat yang dimasukkan pengguna merupakan palindrom atau tidak. Program menggunakan struktur data stack dari library ``<stack>`` untuk membalikkan setengah bagian dari kalimat dan membandingkannya dengan setengah bagian lainnya. Pertama, pengguna diminta untuk memasukkan kalimat. Selanjutnya, fungsi ``isPalindrome(string kalimat)`` memeriksa apakah kalimat tersebut merupakan palindrom atau tidak dengan cara membandingkan setengah bagian pertama kalimat dengan setengah bagian kedua kalimat yang telah dibalik. Hasil pengecekan palindrom kemudian ditampilkan kepada pengguna melalui output yang sesuai.

2. Unguided 2

B. Source code

```
#include <iostream>
#include <stack>
#include <string>

using namespace std;

string reverseLetters(string sentence) {
    stack<char> letters;
    string reversedSentence = "";

    for (char letter : sentence) {
        letters.push(letter);
    }

    while (!letters.empty()) {
        reversedSentence += letters.top();
        letters.pop();
    }

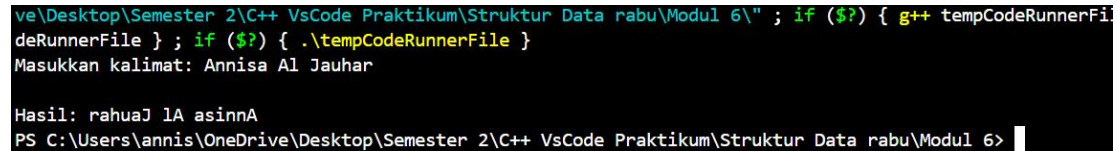
    return reversedSentence;
}

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

    string reversedSentence = reverseLetters(sentence);
    cout << "\nHasil: " << reversedSentence << endl;

    return 0;
}
```

Screenshoot program



```
ve\Desktop\Semester 2\C++ VsCode Praktikum\Struktur Data rabu\Modul 6\" ; if ($?) { g++ tempCodeRunnerFi
deRunnerFile } ; if ($?) { .\tempCodeRunnerFile }
Masukkan kalimat: Annisa Al Jauhar

Hasil: rahuaJ lA asinnA
PS C:\Users\annis\OneDrive\Desktop\Semester 2\C++ VsCode Praktikum\Struktur Data rabu\Modul 6>
```

Deskripsi program

Program di atas adalah sebuah program yang bertujuan untuk membalikkan urutan huruf dari sebuah kalimat yang dimasukkan oleh pengguna. Program menggunakan struktur data stack dari library '<stack>' untuk menyimpan setiap huruf dari kalimat secara terbalik. Pertama, pengguna diminta untuk memasukkan kalimat melalui 'getline(cin, sentence)'. Selanjutnya, fungsi 'reverseLetters(string sentence)' digunakan untuk membalikkan urutan huruf dari kalimat tersebut. Dalam fungsi ini, setiap huruf dari kalimat dimasukkan ke dalam stack dengan menggunakan loop 'for'. Setelah itu, program mengeluarkan huruf-huruf dari stack dan menyusunnya kembali ke dalam string 'reversedSentence' dengan menggunakan loop 'while'. Hasil dari proses ini adalah kalimat yang huruf-hurufnya telah dibalikkan urutannya. Kalimat yang telah dibalikkan tersebut kemudian ditampilkan kepada pengguna melalui output yang sesuai.

BAB IV

KESIMPULAN

Kesimpulannya, stack adalah struktur data yang mengikuti prinsip First In Last Out (FILO) atau Last In First Out (LIFO), di mana data ditambahkan dan diambil dari ujung atas tumpukan, dan tumpukan memiliki sifat dinamis yang memungkinkan penambahan dan pengambilan data secara fleksibel.