

LAPORAN PRAKTIKUM

MODUL VI STACK



Disusun oleh:
Tegar Bangkit Wijaya
NIM: 2311102027

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.

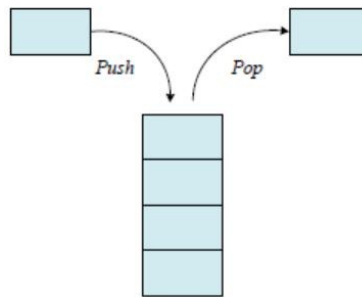
BAB II

DASAR TEORI

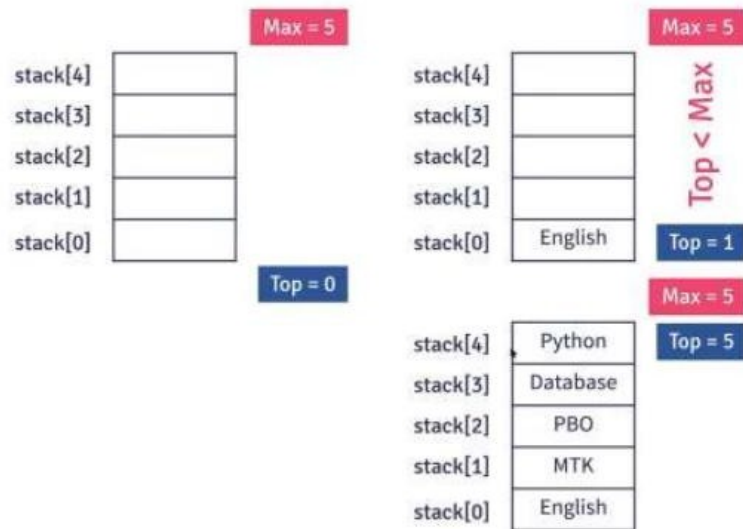
B.DASAR TEORI

a. Pengetian Stack

Stack adalah kumpulan data yang disusun satu di atas yang lain, mirip dengan tumpukan. Stack adalah salah satu struktur data yang mengikuti prinsip LIFO (Last In First Out), yang berarti elemen yang terakhir dimasukkan ke dalam stack akan menjadi elemen pertama yang diambil. Untuk menambahkan elemen ke bagian atas stack, dilakukan operasi yang disebut push. Sementara itu, untuk menghapus elemen dari bagian atas stack, digunakan operasi yang disebut 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 padaposisi paling atas atau ujung.
- Pop (Keluarkan):** Menghapus elemen dari posisi paling atas atau ujungtumpukan
- Top (Atas):** Mendapatkan nilai atau melihat elemen teratas padatumpukan 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.

- c. Peek (Lihat): Melihat nilai atau elemen pada posisi tertentu dalam tumpukan tanpa menghapusnya.
- d. Clear (Hapus Semua): Mengosongkan atau menghapus semua elemen dari tumpukan.
- e. 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

```
PS C:\praktikum struktur data\Modul6> cd "c:\praktikum struktur data\Modul6\" ; if ($?) { g++ guided1.cpp -o guided1 } ; if ($?) { .\guided1 }
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:\praktikum struktur data\Modul6>
```

Deskripsi program

Program tersebut adalah implementasi dari stack menggunakan array, di mana stack digunakan untuk menyimpan judul buku dengan prinsip Last In, First Out (LIFO). Array `arrayBuku` berfungsi sebagai penyimpan data dengan kapasitas maksimal 5, sedangkan variabel `top` menunjukkan posisi teratas dari stack. Fungsi utilitas seperti `isFull()` dan `isEmpty()` digunakan untuk memeriksa kondisi stack. Operasi dasar stack seperti `pushArrayBuku()` untuk menambah data, `popArrayBuku()` untuk menghapus data, `peekArrayBuku()` untuk melihat data pada posisi tertentu, dan `changeArrayBuku()` untuk mengubah data diimplementasikan dalam program ini. Program diuji melalui fungsi `main()`, yang melakukan berbagai operasi pada stack untuk menunjukkan bagaimana stack dapat digunakan secara efisien untuk menyimpan dan mengelola data, khususnya judul buku.

LATIHAN KELAS - UNGUIDED

1. Unguided 1

Source code

```
// Font Courier New (10)
```

Screenshoot program

Deskripsi program

2. Unguided 2

1. Unguided 1

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);
```

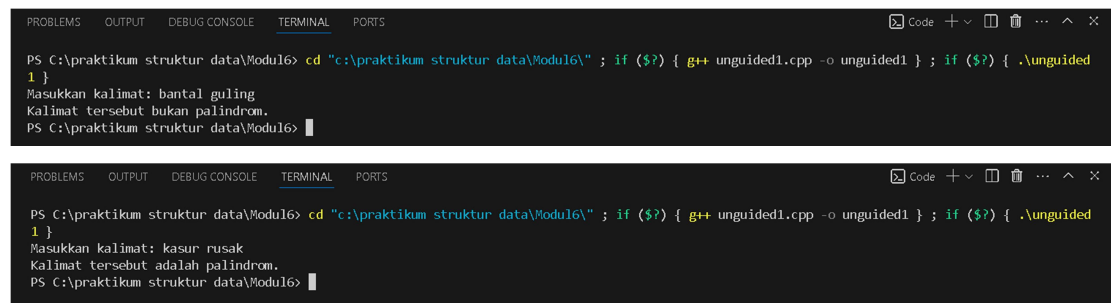
```

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

        return 0;
    }
}

```

Screenshoot program



```

PS C:\praktikum struktur data\Modul6> cd "c:\praktikum struktur data\Modul6\" ; if ($?) { g++ unguided1.cpp -o unguided1 } ; if ($?) { .\unguided1 }
Masukkan kalimat: bantal guling
Kalimat tersebut bukan palindrom.
PS C:\praktikum struktur data\Modul6>

PS C:\praktikum struktur data\Modul6> cd "c:\praktikum struktur data\Modul6\" ; if ($?) { g++ unguided1.cpp -o unguided1 } ; if ($?) { .\unguided1 }
Masukkan kalimat: kasur rusak
Kalimat tersebut adalah palindrom.
PS C:\praktikum struktur data\Modul6>

```

Deskripsi program

Program ini dirancang untuk memeriksa apakah kalimat yang dimasukkan oleh pengguna adalah palindrom atau tidak. Program ini menggunakan struktur data stack dari library '<stack>' untuk membalikkan setengah bagian dari kalimat dan membandingkannya dengan setengah bagian lainnya. Langkah-langkah yang dilakukan adalah sebagai berikut:

1. Meminta Input dari Pengguna: Pengguna diminta untuk memasukkan sebuah kalimat.
2. Fungsi 'isPalindrome(string kalimat)': Fungsi ini memeriksa apakah kalimat tersebut merupakan palindrom dengan membandingkan setengah bagian pertama dari kalimat dengan setengah bagian kedua yang sudah dibalik menggunakan stack.

3. Menampilkan Hasil: Hasil dari pengecekan palindrom ditampilkan kepada pengguna melalui output yang sesuai.

2. Unguided2

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;
}

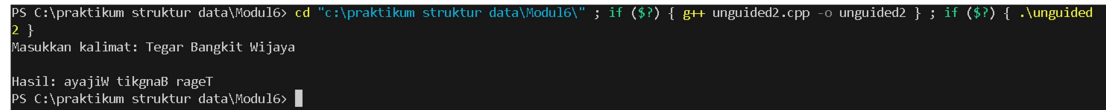
int main() {
    string sentence;
    cout << "Masukkan kalimat:
    ";getline(cin, sentence);

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

```
        reversedSentence << endl;

        return 0;
    }
```

Screenshoot program



```
PS C:\praktikum struktur data\Modul6> cd "c:\praktikum struktur data\Modul6\" ; if ($?) { g++ unguided2.cpp -o unguided2 } ; if ($?) { .\unguided2 }
Masukkan kalimat: Tegar Bangkit Wijaya
Hasil: ayajjIW tikgnaB rageT
PS C:\praktikum struktur data\Modul6>
```

Deskripsi program

Program ini bertujuan untuk membalikkan urutan huruf dalam sebuah kalimat yang dimasukkan oleh pengguna. Program menggunakan struktur data stack dari library ``<stack>`` untuk menyimpan setiap huruf dari kalimat secara terbalik. Langkah-langkah yang dilakukan oleh program adalah sebagai berikut:

1. Meminta Input dari Pengguna: Pengguna diminta untuk memasukkan sebuah kalimat melalui ``getline(cin, sentence)``.
2. Fungsi ``reverseLetters(string sentence)``: Fungsi ini digunakan untuk membalikkan urutan huruf dalam kalimat. Dalam fungsi ini, setiap huruf dari kalimat dimasukkan ke dalam stack menggunakan loop ``for``.
3. Menyusun Kembali Kalimat Terbalik: Setelah semua huruf dimasukkan ke dalam stack, program mengeluarkan huruf-huruf tersebut dari stack dan menyusunnya kembali ke dalam string ``reversedSentence`` menggunakan loop ``while``.
4. Menampilkan Hasil: Kalimat yang telah dibalik urutannya kemudian ditampilkan kepada pengguna melalui output yang sesuai.

BAB IV

KESIMPULAN

Secara keseluruhan, stack adalah sebuah struktur data yang mengikuti prinsip Last In First Out (LIFO), yang berarti data terakhir yang dimasukkan akan menjadi data pertama yang diambil, dan sebaliknya. Operasi penambahan data (push) dan pengambilan data (pop) dilakukan dari ujung atas tumpukan. Stack juga memiliki sifat dinamis yang memungkinkan penambahan dan pengambilan data secara fleksibel, tanpa memerlukan alokasi memori yang besar. Dengan demikian, stack sangat berguna dalam berbagai aplikasi yang membutuhkan penyimpanan dan pengelolaan data dengan prinsip LIFO, seperti dalam implementasi algoritma, manajemen memori, dan banyak lagi.