

LAPORAN PRAKTIKUM

MODUL 6 STACK



Disusun oleh:
Denny Budiansyach
NIM: 2311102022

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

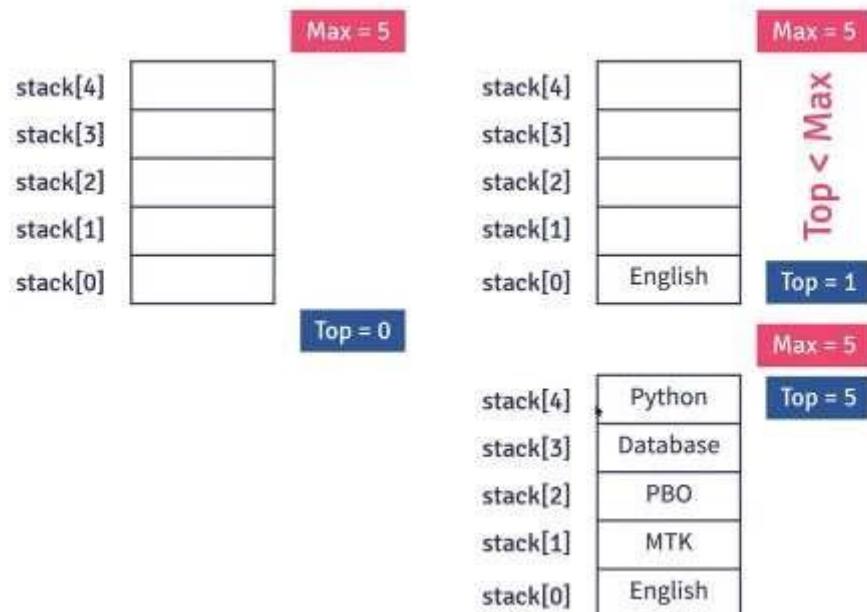
1. Praktikan dapat memahami konsep stack pada struktur data dan algoritma
2. Praktikan dapat mengimplementasikan operasi operasi pada stack
3. Praktikan dapat memecahkan permasalahan dengan solusi stack

BAB II

DASAR TEORI

Stack adalah struktur data sederhana yang digunakan untuk menyimpan data (mirip dengan Linked Lists). Dalam tumpukan, urutan kedatangan data penting. Sebuah tumpukan piring di kafetaria adalah contoh bagus dari tumpukan. Piring ditambahkan ke tumpukan saat mereka dibersihkan dan ditempatkan di bagian atas. Ketika sebuah piring dibutuhkan, diambil dari bagian atas tumpukan. Piring pertama yang ditempatkan di tumpukan adalah yang terakhir digunakan.

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:

- a. Push (Masukkan): Menambahkan elemen ke dalam tumpukan pada posisi paling atas atau ujung.
 - b. Pop (Keluarkan): Menghapus elemen dari posisi paling atas atau ujung tumpukan.
 - c. Top (Atas): Mendapatkan nilai atau melihat elemen teratas pada tumpukan tanpa menghapusnya.
 - d. 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.
- 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

```

Deskripsi program

Program di atas adalah bentuk implementasi dari stack yang menggunakan array untuk menyimpan dan mengelola koleksi buku. Dengan menerapkan prinsip stack maka setiap data akan disimpan pada program dengan metode LIFO (Last In, First Out), di mana data terakhir yang dimasukkan adalah data pertama yang dapat diambil. Program ini menerapkan beberapa operasi dasar untuk mengelola stack,

seperti penambahan (push), penghapusan (pop), pengubahan, pengaksesan elemen tertentu (peek), dan pemeriksaan status stack. Setiap operasi dasar tersebut dijalankan oleh fungsi fungsi berikut:

1. isFull():

Melakukan pernyataan apakah stack sudah penuh dengan memeriksa posisi data terakhir/tertinggi dalam stack sama dengan jumlah maksimal data yang dapat ditampung oleh stack. Fungsi akan mengembalikan `true` jika stack penuh, dan `false` jika tidak.

2. isEmpty():

Memeriksa apakah stack kosong dengan memeriksa index top pada stack. Fungsi akan mengembalikan `true` jika stack kosong, dan `false` jika tidak.

3. pushArrayBuku(string data):

Menambahkan setiap data buku baru ke array stack pada posisi TOP atau posisi teratas. Jika stack penuh, fungsi akan menampilkan pesan bahwa data telah penuh.

4. popArrayBuku():

Menghapus buku dari stack dengan cara menghapus data pada posisi TOP atau posisi teratas pada stack. Jika stack kosong, fungsi akan menampilkan pesan bahwa tidak ada data yang dihapus.

5. peekArrayBuku(int posisi):

Menampilkan buku pada posisi tertentu dari atas stack. Jika stack kosong, fungsi akan menampilkan pesan bahwa tidak ada data yang bisa dilihat.

6. countStack():

Menampilkan jumlah buku yang ada dalam stack.

7. `changeArrayBuku(int posisi, string data):`

Mengubah data buku pada posisi tertentu dengan cara mengakses seluruh index pada array stack mulai dari posisi top kebawah sampai pada index posisi data yang akan diubah lalu mengganti value dari index tersebut dengan data yang baru. Jika posisi melebihi jumlah buku dalam stack, menampilkan pesan bahwa posisi melebihi data yang ada.

8. `destroyArraybuku():`

Menghapus semua data buku dari stack dan mengosongkan stack.

9. `cetakArrayBuku():`

Mencetak semua buku yang ada dalam stack dari atas ke bawah. Jika stack kosong, menampilkan pesan bahwa tidak ada data yang dicetak.

LATIHAN KELAS - UNGUIDED

UNGUIDED 1

1. Unguided Source code

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

char stackData[77];
int top = -1;

bool isFull(int n) {
    return (top == n - 1);
}
bool isEmpty() {
    return (top == -1);
}
void push(char x) {
    if (!isFull(77)) {
        stackData[++top] = x;
    }
}
char pop() {
    if (isEmpty())
    {
        cout << "Tidak ada data yang dihapus" << endl;
    }
    else
    {
        return stackData[top--];
    }
}
bool Palindrom(string str) {
    int n = str.length();
    for (int i = 0; i < n; i++) {
        push(str[i]);
    }
    string rvrsdStr = "";
    while (!isEmpty()) {
        rvrsdStr += pop();
    }
    return (str == rvrsdStr);
}
```

```

}

int main() {
    string str;
    cout << "Kalimat : ";
    getline(cin, str);
    if (Palindrom(str)) {
        cout << "Kalimat "<<str<<" adalah Palindrom" << endl;
    } else {
        cout << "Kalimat "<<str<<" bukan Palindrom"<< endl;
    }
    return 0;
}

```

Screenshoot program

```

Kalimat : telolet
Kalimat telolet adalah Palindrom
PS C:\Users\rizqi\Documents\COLLEGE\SEMESTER II\PRAKTIKUM STRUKDAT\
● M STRUKDAT\Mod 6 - Stack\" ; if ($?) { g++ UnGuided1.cpp -o UnGuide
Kalimat : tolelet
Kalimat tolelet bukan Palindrom

```

Deskripsi program

Program di atas adalah bentuk implementasi dari stack yang menggunakan array untuk memeriksa sebuah kata termasuk dalam jenis kata palindrom atau bukan. Dengan menggunakan metode lifo, program akan menyimpan setiap karakter huruf pada kata ke dalam array stackData. Setelah semua karakter tersimpan program kemudian akan menghapus karakter pada array stackData dan memindahkan karakter tersebut kedalam array rvrsdStr secara berurutan dari data teratas/terakhir hingga data pertama pada stackData, iterasi ini akan terus dilakukan hingga seluruh karakter pada array stackData terhapus dan dipindahkan.

Dengan cara tersebut maka seluruh karakter pada stackData akan tersimpan kembali pada array rvrsdStr dengan urutan terbalik. Untuk menentukan apakah kata palindrom atau tidak program akan membandingkan karakter pada stackData dengan karakter yang telah di reverse dalam rvrsdStr, jika hasilnya sama maka kata tersebut adalah Palindrom, jika ternyata berbeda maka kata tersebut bukan termasuk kata Palindrom.

UNGUIDED 2

2. Unguided Source code

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

char stackData[77];
int top = -1;

bool isFull(int n) {
    return (top == n - 1);
}

bool isEmpty() {
    return (top == -1);
}

void push(char x) {
    if (!isFull(77)) {
        stackData[++top] = x;
    }
}

char pop() {
    return stackData[top--];
}

bool reverse(string str) {
    int n = str.length();
    for (int i = 0; i < n; i++) {
        push(str[i]);
    }
    string rvrsdStr = "";
    while (!isEmpty()) {
        rvrsdStr += pop();
    }
    cout<< "String D Reverse : "<<rvrsdStr;
}

int main() {
    string kalimat;
    cout << "String          : ";
    getline(cin, kalimat);
    reverse(kalimat);
    return 0;
}
```

Screenshoot program

```
M STRUKDAT\Mod 6 - Stack\" ; if ($?) { g++ UnGuided2.cpp -o UnGuided2
String      : Institut Teknologi Telkom Purwokerto
String D Reverse : otrekowruP mokleT igolonkeT tutitsnI
```

Deskripsi program

Program di atas merupakan program yang cara kerjanya sama dengan program Unguided 1. Jika pada Unguided 1 `rvrsdStr` dibandingkan dengan `stackData` untuk mengidentifikasi Palindrom, maka pada program ini `rvrsdStr` hanya perlu ditampilkan. Program ini menerapkan beberapa fungsi operasi dasar untuk melakukan reverse kalimat, seperti penambahan (`push`), penghapusan (`pop`), dan pemeriksaan status stack. Setiap operasi dasar tersebut dijalankan oleh fungsi fungsi berikut:

1. `isFull(int n)`:

Melakukan pernyataan apakah stack sudah penuh dengan memeriksa posisi data terakhir/tertinggi dalam stack sama dengan jumlah maksimal data yang dapat ditampung oleh stack. Fungsi akan mengembalikan `'true'` jika stack penuh, dan `'false'` jika tidak.

2. `isEmpty()`:

Memeriksa apakah stack kosong dengan memeriksa index top pada stack. Fungsi akan mengembalikan `'true'` jika stack kosong, dan `'false'` jika tidak.

3. `push(char x)`:

Fungsi ini akan menambahkan setiap karakter pada string ke dalam array `stackData` dengan mengisinya pada posisi TOP atau posisi teratas. Iterasi tersebut akan terus diulangi hingga status stack adalah full yang ditandai dengan fungsi `isFull(int n)` mereturn nilai `true`.

4. pop ():

Fungsi ini akan menghapus karakter yang tersimpan pada stackData dengan cara menghapus data pada posisi TOP atau posisi teratas pada stack.

5. reverse (string str):

Fungsi ini adalah fungsi akan melakukan reverse string dengan memanfaatkan fungsi fungsi sebelumnya. Fungsi reverse akan mengidentifikasi panjang atau jumlah karakter pada string input dengan str.length. jumlah karakter tersebut kemudian akan menjadi inisialisasi nilai pada variabel int n. Lalu program akan menggunakan fungsi push pada setiap karakter pada array string (str[i]) ke dalam array stackData, dengan begitu setiap karakter string inputan akan tersimpan di index index stackData.

Kemudian untuk melakukan reverse string program akan menghapus karakter pada array stackData kemudian menyimpannya pada variabel rvrsdStr. Penghapusan sekaligus pemindahan karakter ini akan dilakukan berurutan dari data teratas(top) hingga data terbawah pada stackData, dengan memindahkannya dari atas kebawah maka rvrsdStr akan menyimpan seluruh karakter stackData secara terbalik. Pada akhir fungsi akan melakukan cout dari rvrsdStr untuk mencetak string yang sudah di balik/reverse.

BAB IV

KESIMPULAN

Stack adalah struktur penyimpanan data yang berbentuk linier dimana elemen terakhir yang masuk kedalam stack adalah elemen pertama yang dapat diakses atau dikeluarkan oleh stack. Prinsip ini disebut dengan Last In First Out atau LIFO. Analogi untuk prinsip ini adalah seperti kaleng tabung bola tenis, bola tenis yang dapat diambil dalam kaleng adalah bola teratas sekaligus bola terakhir yang dimasukkan dalam kaleng. Prinsip ini sama dengan LIFO dimana bola terakhir yang dimasukkan ke dalam kaleng adalah bola pertama yang dapat diambil atau dikeluarkan.

DAFTAR PUSTAKA

- GeeksforGeeks*. (2022, Juli 27). Diambil kembali dari [geeksforgeeks.com](https://www.geeksforgeeks.org/fifo-vs-lifo-approach-in-programming/):
[https://www.geeksforgeeks.org/fifo-vs-lifo-approach-in-programming//](https://www.geeksforgeeks.org/fifo-vs-lifo-approach-in-programming/)
- Asisten Praktikum. (2024). *MODUL 6 STACK*, Learning Managament System.