

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL VI
STACK**



Disusun Oleh :

NAMA : GALIH TRISNA
NIM : 2311102050

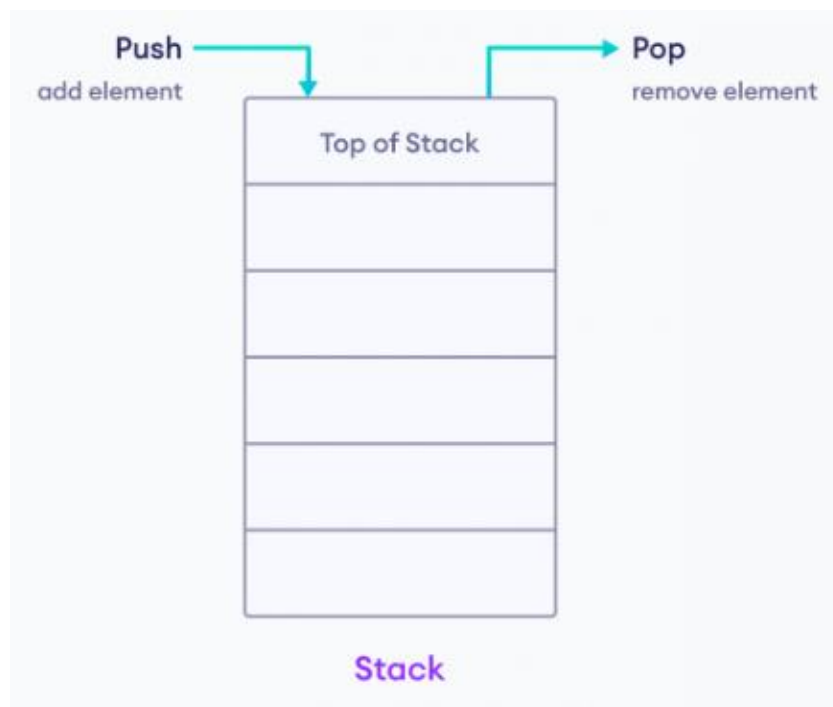
Dosen

Wahyu Andi Saputra, S.Pd., M.Eng

**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFROMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
2024**

A. Dasar Teori

Stack adalah struktur data mendasar lainnya yang sering digunakan untuk memecahkan masalah pemrograman. Mirip dengan queue, ini digunakan ketika kita membutuhkan sebuah wadah yang membatasi urutan objek yang diakses. Perbedaan utamanya adalah tumpukan menggunakan prinsip sebaliknya. Jika queue berdasarkan prinsip masuk pertama, keluar pertama (FIFO), stack adalah masuk terakhir, keluar pertama (LIFO) Artinya, objek dalam stack diakses dalam urutan berlawanan saat ditambahkan. Objek yang terakhir ditambahkan akan diakses terlebih dahulu. Stack sering kali divisualisasikan sebagai menara objek vertikal, dengan item yang paling baru ditambahkan berada di bagian atas. Dengan demikian, hanya objek paling atas yang dapat diakses:



Di C++, class Stack menyediakan berbagai metode untuk melakukan operasi berbeda pada tumpukan :

1. `push()` tambahkan elemen ke dalam tumpukan
2. `pop()` Menghapus elemen dari tumpukan
3. `top()` mengembalikan elemen di bagian atas tumpukan
4. `size()` mengembalikan jumlah elemen dalam tumpukan
5. `empty()` mengembalikan nilai true jika tumpukan kosong

B. Guided

Guided 1

```
#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())

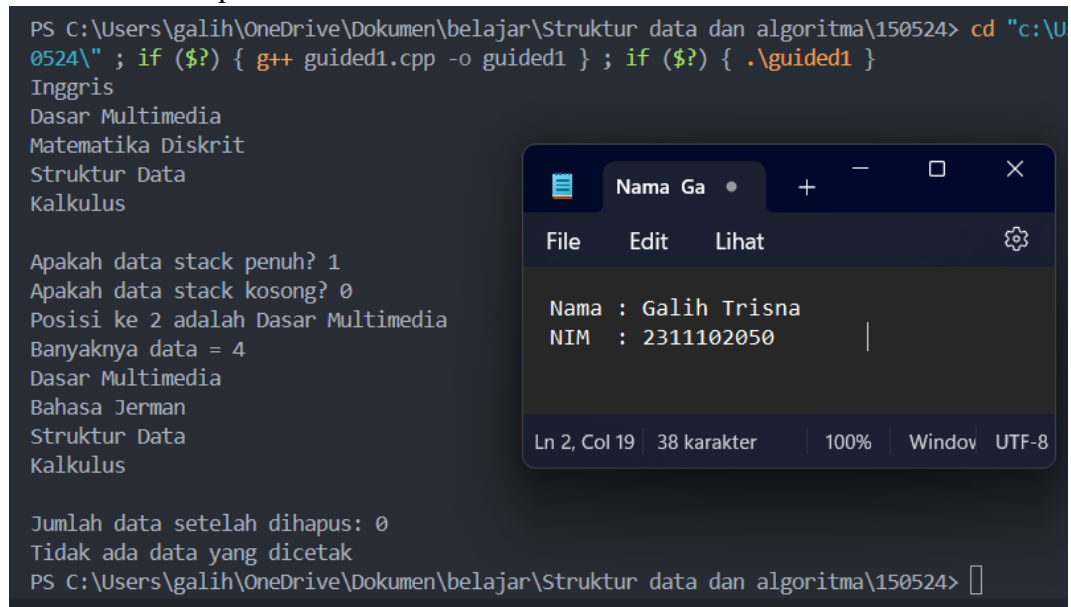
```

```

    {
        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;
}

```

Screenshots Output



The screenshot shows a terminal window with the following output:

```
PS C:\Users\galih\OneDrive\Dokumen\belajar\Struktur data dan algoritma\150524> cd "c:\U
0524\" ; 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:\Users\galih\OneDrive\Dokumen\belajar\Struktur data dan algoritma\150524>
```

Overlaid on the terminal is a Notepad++ window titled 'Nama Ga'. It contains the following text:

```
Nama : Galih Trisna
NIM : 2311102050
```

The Notepad++ status bar at the bottom indicates 'Ln 2, Col 19 | 38 karakter | 100% | Window | UTF-8'.

Deskripsi:

Kode ini mengimplementasikan stack menggunakan array untuk menyimpan buku. Stack adalah struktur data yang mengikuti prinsip LIFO (Last In, First Out), di mana elemen terakhir yang dimasukkan adalah yang pertama diambil. string arrayBuku[5]; mendeklarasikan array berukuran 5 untuk menyimpan string buku. int maksimal = 5, top = 0; mendefinisikan kapasitas maksimum stack (5) dan indeks teratas top yang mulai dari 0. Kemudian ada beberapa fungsi di program tersebut yaitu :

1. Fungsi isEmpty:
isEmpty() memeriksa apakah stack kosong dengan memeriksa apakah top sama dengan 0.
2. Fungsi isFull:
isFull() memeriksa apakah stack penuh dengan membandingkan top dengan maksimal.
3. Fungsi pushArrayBuku:
Menambahkan buku ke stack jika stack belum penuh, dengan menempatkan buku di posisi top dan menaikkan top.
4. Fungsi popArrayBuku:
Menghapus buku dari stack jika stack tidak kosong, dengan mengosongkan posisi teratas dan menurunkan top.

5. Fungsi peekArrayBuku:
Menampilkan buku pada posisi tertentu dari atas stack, dengan menghitung indeks yang sesuai.
6. Fungsi countStack:
Mengembalikan jumlah buku yang ada di stack (nilai top).
7. Fungsi changeArrayBuku:
Mengganti buku pada posisi tertentu dari atas stack dengan buku baru, dengan menghitung indeks yang sesuai.
8. Fungsi destroyArraybuku:
Mengosongkan seluruh stack dengan mengosongkan setiap elemen array dan mengatur top ke 0.
9. Fungsi cetakArrayBuku:
Mencetak semua buku dalam stack dari atas ke bawah.
10. Fungsi main:
Menambahkan beberapa buku ke stack menggunakan pushArrayBuku.
Mencetak isi stack.
Memeriksa apakah stack penuh atau kosong.
Melihat buku pada posisi tertentu dari atas stack.
Menghapus buku dari stack dengan popArrayBuku.
Menghitung dan mencetak jumlah buku dalam stack.
Mengganti buku pada posisi tertentu.
Mencetak isi stack setelah perubahan.
Mengosongkan stack dan mencetak jumlah buku setelah dihapus.

C. Unguided/Tugas

Unguided 1

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

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

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

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

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

void popChar() {
    if (isEmpty()) {
        cout << "Stack kosong" << endl;
    } else {
        top--;
    }
}

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



```

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

    int lengthInput = input.length();
    for (int i = 0; i < lengthInput; ++i) {
        pushChar(input[i]);
    }

    bool palindrom = true;
    for (int i = 0; i < lengthInput; ++i) {
        if (input[i] != peekChar()) {
            palindrom = false;
            break;
        }
        popChar();
    }

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

    return 0;
}

```

Screenshots Output

- Jika palindrom

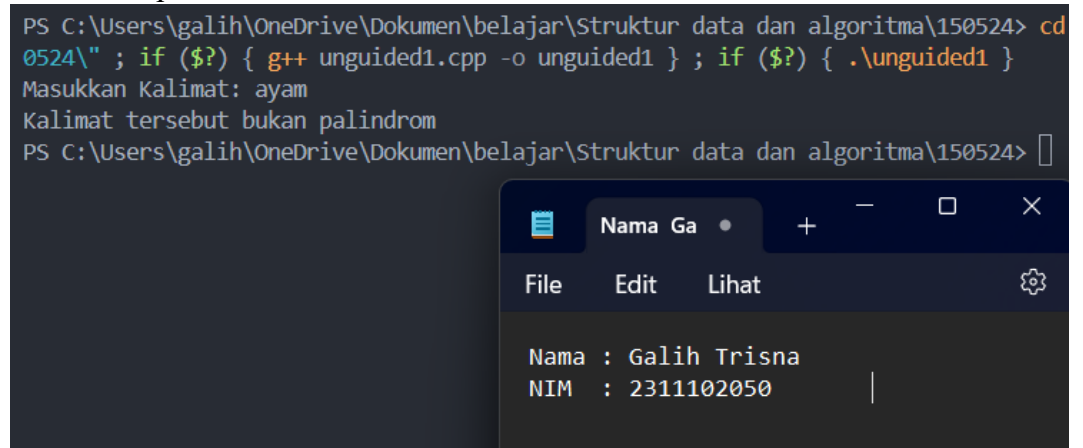
```

PS C:\Users\galih\OneDrive\Dokumen\belajar\Struktur data dan algoritma\150524> c
0524\" ; if ($?) { g++ unguided1.cpp -o unguided1 } ; if ($?) { .\unguided1 }
Masukkan Kalimat: kakak
Kalimat tersebut adalah palindrom
PS C:\Users\galih\OneDrive\Dokumen\belajar\Struktur data dan algoritma\150524>

```

Nama Ga • + - □ ×
File Edit Lihat ⚙️
Nama : Galih Trisna
NIM : 2311102050 |

- Jika bukan palindrom



```

PS C:\Users\galih\OneDrive\Dokumen\belajar\Struktur data dan algoritma\150524> cd 0524\
; if ($?) { g++ unguided1.cpp -o unguided1 } ; if ($?) { .\unguided1 }
Masukkan Kalimat: ayam
Kalimat tersebut bukan palindrom
PS C:\Users\galih\OneDrive\Dokumen\belajar\Struktur data dan algoritma\150524>

```

Nama : Galih Trisna
NIM : 2311102050

Deskripsi

Program diatas merupakan program yang dapat mengecek apakah sebuah kata atau kalimat ini adalah palindrom atau bukan. Pada awalnya program mendeklarasikan array sebagai stack yaitu charArray dan juga mendeklarasikan maksimal dan top. Dalam program ini pengguna diminta untuk memasukan sebuah string. Kemudian setelah di inputkan, program tersebut akan memasukan string tersebut ke dalam stack menggunakan fungsi pushChar. Kemudian program tersebut mengecek palindrom dengan cara membandingkan setiap karakter yang di inputkan dengan karakter yang ada di posisi paling atas dalam stack menggunakan fungsi peekChar. Proses ini berulang sampai karakter yang dibandingkan habis atau ada ditemukan karakter yang berbeda. Jika ditemukan karakter yang berbeda saat proses perbandingan, maka perulangan akan berhenti dan mengubah variable palindrom menjadi bernilai false. Setiap setelah perbandingan terjadi maka karakter puncak stack akan dihapus dengan fungsi popChar. Kemudian jika variable palindrom bernilai true maka program akan menampilkan pesan “Kalimat ini adalah palindrom”, namun jika nilai variable palindrom bernilai false maka akan menampilkan pesan “Kalimat ini bukan palindrom”.

Unguided 2

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

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

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

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

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

void popChar() {
    if (isEmpty()) {
        cout << "Stack kosong" << endl;
    } else {
        charArray[top - 1] = '\0';
        top--;
    }
}

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

int main() {
```

```

string input;
cout << "Masukkan Kalimat: ";
getline(cin, input);

int lengthInput = input.length();
for (int i = 0; i < lengthInput; ++i) {
    pushChar(input[i]);
}

cout << "Hasil: ";
while (!isEmpty()) {
    cout << peekChar();
    popChar();
}

return 0;
}

```

Screenshots Output

```

PS C:\Users\galih\OneDrive\Dokumen\belajar\Struktur data dan algoritma\150524> cd 0524\
; if ($?) { g++ unguided2.cpp -o unguided2 } ; if ($?) { .\unguided2 }
Masukkan Kalimat: Institut Teknologi Telkom Purwokerto
Hasil: otrekowrup moklet igolonket tutitsnI
PS C:\Users\galih\OneDrive\Dokumen\belajar\Struktur data dan algoritma\150524>

```

Nama : Galih Trisna
NIM : 2311102050

Deskripsi

Kode ini mengimplementasikan sebuah stack menggunakan array untuk menyimpan karakter-karakter dari sebuah kalimat yang dimasukkan oleh pengguna. Stack digunakan untuk mencetak karakter-karakter kalimat tersebut dalam urutan terbalik. Kode dimulai dengan mendeklarasikan array karakter berukuran 100 sebagai stack, serta variabel untuk melacak kapasitas maksimum stack dan indeks teratas. Fungsi `isFull` dan `isEmpty` digunakan untuk memeriksa apakah stack penuh atau kosong. Fungsi `pushChar` menambahkan karakter ke dalam stack jika stack belum penuh, sementara `popChar` menghapus karakter dari puncak stack jika stack tidak kosong. Fungsi

peekChar mengembalikan karakter di puncak stack tanpa menghapusnya. Dalam fungsi main, pengguna diminta memasukkan sebuah kalimat menggunakan getline untuk menangkap seluruh baris input. Setiap karakter dari kalimat tersebut kemudian dimasukkan ke dalam stack menggunakan pushChar. Setelah semua karakter dimasukkan, program mencetak karakter-karakter tersebut dalam urutan terbalik dengan menggunakan peekChar dan popChar dalam sebuah loop hingga stack kosong. Proses ini memastikan bahwa karakter pertama yang dimasukkan ke stack adalah yang terakhir dicetak, sehingga menghasilkan urutan karakter yang terbalik dari kalimat asli.

D. Kesimpulan

Stack adalah struktur data LIFO (Last In, First Out) yang memungkinkan elemen terakhir yang dimasukkan menjadi yang pertama diambil. Operasi utamanya adalah `push` (menambahkan elemen), `pop` (menghapus elemen), dan `peek` (melihat elemen teratas tanpa menghapusnya). Stack sering digunakan dalam pengelolaan undo-redo, penelusuran graf, dan evaluasi ekspresi matematika.

E. Referensi

Asisten Praktikum. (2024). Modul VI : Stack

Programiz. (n.d.). Stack. Diakses pada 19 Mei 2024, dari <https://www.programiz.com/cpp-programming/stack>

Introduction to Stacks using std::stack. (n.d.). Stack. Diakses pada 19 Mei 2024, dari <https://www.studyplan.dev/pro-cpp/stacks>

IsmyNR. (2018). Contoh Program Stack C++ beserta Penjelasan dan Implementasinya (Tumpukan). Diakses pada 19 Mei 2024, dari <https://www.ismynr.xyz/2018/10/contoh-program-stack-c-penjelasan.html>