

**LAPORAN PRAKTIKUM STRUKTUR  
DATA DAN ALGORITMA**

**MODUL 6  
STACK**



**Disusun Oleh :**  
**Muhammad Hamzah Haifan Ma'ruf**  
**231102091**

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

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

## A. Dasar Teori

Stack adalah salah satu struktur data yang digunakan untuk menyimpan sekumpulan objek ataupun variabel. Sesuai namanya yaitu stack, tidak heran apabila objek yang terkumpul terlihat seperti tumpukan. Nah, karakteristik stack sendiri bersifat LIFO (last in first out). Artinya, data yang terakhir masuk merupakan data yang akan keluar terlebih dahulu. Seperti halnya tumpukan pada umumnya, misalnya tumpukan buku, yang di atas atau yang terakhir masuk harus dikeluarkan terlebih dahulu untuk mendapatkan buku yang berada di tumpukan bawah. 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.

### Kelebihan dan Kekurangan Menggunakan Stack

#### 1. Kelebihan

- Membantu mengelola data dengan metode LIFO
- Secara otomatis membersihkan objek
- Tidak mudah rusak
- Ukuran variabel tidak dapat diubah
- Mengontrol memori secara mandiri

#### 2. Kekurangan

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

### Jenis-Jenis Operasi Stack

#### 1. Pop

Pop pada stack adalah operasi yang berfokus pada penghapusan elemen. Dikarenakan dalam stack programmer hanya memiliki akses pada bagian atas, hanya ada satu elemen yang dapat dihapus.

#### 2. Push

Kebalikan dari pop, operasi Push justru lebih berfokus pada memasukkan elemen ke dalam stack atau tumpukan. Seperti halnya sistem LIFO, programmer atau pengguna hanya dapat memasukkan elemen baru di bagian atas tumpukan.

#### 3. isFull

Operasi stack yang satu ini adalah untuk mengetahui apakah tumpukan sudah penuh atau belum.

#### 4. isEmpty

Kebalikan dari isFull, isEmpty merupakan operasi yang digunakan untuk memeriksa apakah tumpukan kosong atau tidak.

#### 5. Peek

Peek atau mengintip adalah operasi yang dilakukan untuk mengetahui data teratas dari tumpukan tanpa harus menghapusnya.

## B. Guided

### 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;
    }

    current = current->next;
}

// Traversal
void traverse() {
    for (int i = 0; i < MAX_SIZE; i++) {
        Node* current = table[i];
        while (current != nullptr) {
            cout << current->key << ": " << current->value << endl;
            current = current->next;
        }
    }
};

int main() {
    HashTable ht;

    // Insertion
    ht.insert(1, 10);
    ht.insert(2, 20);
    ht.insert(3, 30);

    // Searching
    cout << "Get key 1: " << ht.get(1) << endl;
    cout << "Get key 4: " << ht.get(4) << endl;
    cout << "Get key 2: " << ht.get(2) << endl;
    cout << "Get key 5: " << ht.get(5) << endl;

    // Deletion
    ht.remove(4);

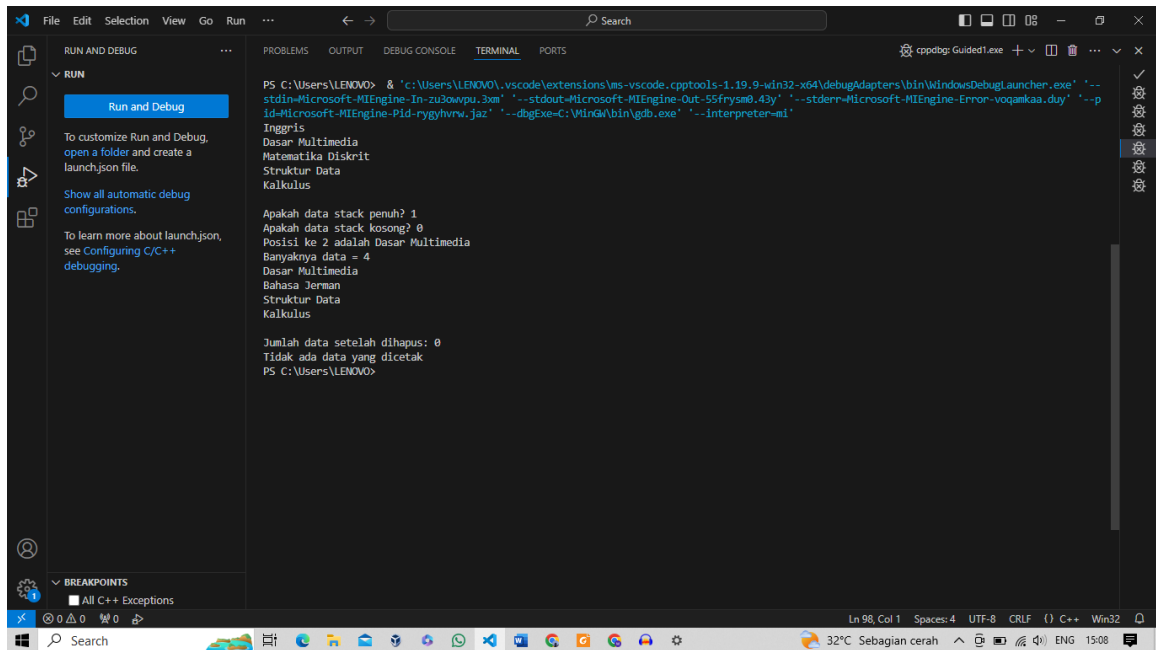
    // Traversal
    ht.traverse();

    return 0;
}

```

```
}
```

## Screenshots Output :



The screenshot shows the Visual Studio Code interface with the terminal window open. The terminal displays the output of a C++ program. The program lists a list of subjects: Inggris, Dasar Multimedia, Matematika Diskrit, Struktur Data, and Kalkulus. It then asks if the data stack is full (1), if it's empty (0), and the position of the 2nd element (2). It also shows the number of data elements (4) and the subjects: Dasar Multimedia, Bahasa Jerman, Struktur Data, and Kalkulus. Finally, it shows the number of data elements after deletion (0) and that there is no data to print.

```
PS C:\Users\LENOVO> & 'c:\Users\LENOVO\.vscode\extensions\ms-vscode.cpptools-1.19.9-win32-x64\debug\adapters\bin\WindowsDebugLauncher.exe' '--
stdin-Microsoft-MIEngine-In-zu6owpu.3m' '--stdout-Microsoft-MIEngine-Out-55frysm0.43y' '--stderr-Microsoft-MIEngine-Error-voqamkaa.duy' '--p
1d-Microsoft-MIEngine-Pid-rygyhwru.jaz' '--dbgExe=c:\Windows\bin\gdb.exe' '--interpreter=mi'

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\LENOVO>
```

## Deskripsi :

Program ini menggunakan array untuk menyimpan data buku, program ini menyediakan fungsi-fungsi dasar seperti penambahan data, penghapusan data, pengubahan data, serta operasi lainnya seperti melihat jumlah data dalam tumpukan dan mencetak isi tumpukan. Program ini membantu pengguna dalam mengelola tumpukan buku dengan mudah, yang merupakan salah satu konsep penting dalam pemrograman.

## C. Unguided

### Unguided 1

#### Source Code

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

//menghapus karakter non-alfabet dari string
```

```

string removeNonAlphanumeric(string str) {
    string result = "";
    //ubah huruf menjadi lowercase
    for (char c : str) {
        if ((c >= 'a' && c <= 'z') || (c >= 'A' && c <= 'Z')) {
            result += tolower(c);
        }
    }
    return result;
}

//menentukan apakah string adalah palindrom atau tidak
bool isPalindrome(string str) {
    int left_091 = 0;
    int right = str.length() - 1;
    while (left_091 < right) {
        //lewati karakter non-alfabet di kiri
        while (left_091 < right && !isalpha(str[left_091])) {
            left_091++;
        }
        //lewati karakter non-alfabet di kanan
        while (left_091 < right && !isalpha(str[right])) {
            right--;
        }
        //periksa apakah karakter di kiri sama dengan karakter di kanan
        if (tolower(str[left_091]) != tolower(str[right])) {
            return false;
        }
        left_091++;
        right--;
    }
    return true;
}

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

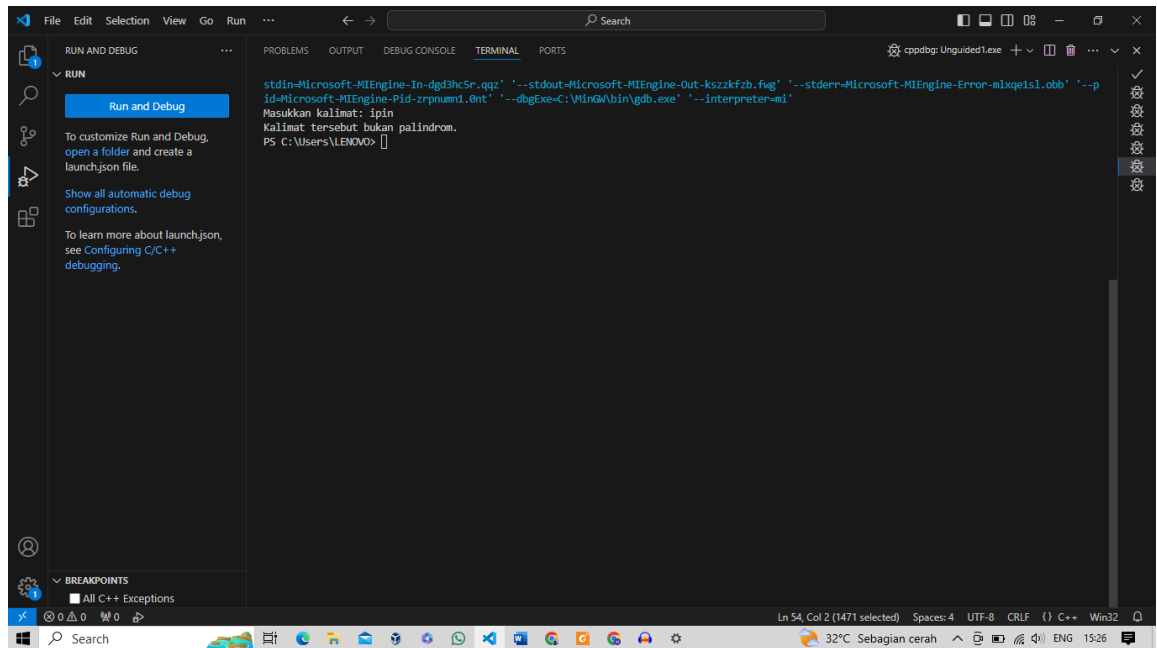
    string cleanedString = removeNonAlphanumeric(kalimat);

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

    return 0;
}

```

## Screenshots Output :



## Deskripsi :

Program ini terdiri dari dua fungsi utama `removenonalphnumeric` dan `isPalindrome`. Pertama, `removenonalphnumeric` bertugas menghapus semua karakter non-alfabet dari sebuah string. Dalam prosesnya, fungsi ini iterasi melalui setiap karakter dalam string input, menyaring hanya huruf (baik huruf besar maupun kecil), dan mengabaikan karakter lainnya. Selanjutnya, huruf-huruf besar diubah menjadi huruf kecil menggunakan fungsi `tolower`, sehingga menghasilkan sebuah string yang bersih dari karakter non-alfabet dan sudah dalam bentuk huruf kecil.

Kedua, fungsi `isPalindrome` digunakan untuk menentukan apakah sebuah string adalah palindrom atau tidak. Algoritma ini memanfaatkan pendekatan dua penunjuk (`left` dan `right`) yang bergerak dari ujung-ke-tengah string, melewati karakter non-alfabet, dan membandingkan karakter-karakter yang relevan. Jika ada perbedaan antara pasangan karakter yang relevan dari kedua sisi string, maka string tersebut tidak dapat menjadi palindrom. Namun, jika tidak ada perbedaan, maka string tersebut adalah palindrom. Di dalam fungsi main, pengguna diminta untuk memasukkan sebuah kalimat. Kalimat tersebut kemudian dibersihkan dari karakter non-alfabet dengan menggunakan `removenonalphnumeric`, lalu dicek apakah merupakan palindrom atau tidak dengan menggunakan `isPalindrome`. Hasilnya akan ditampilkan kepada pengguna.



## Unguided 2

### Source Code

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

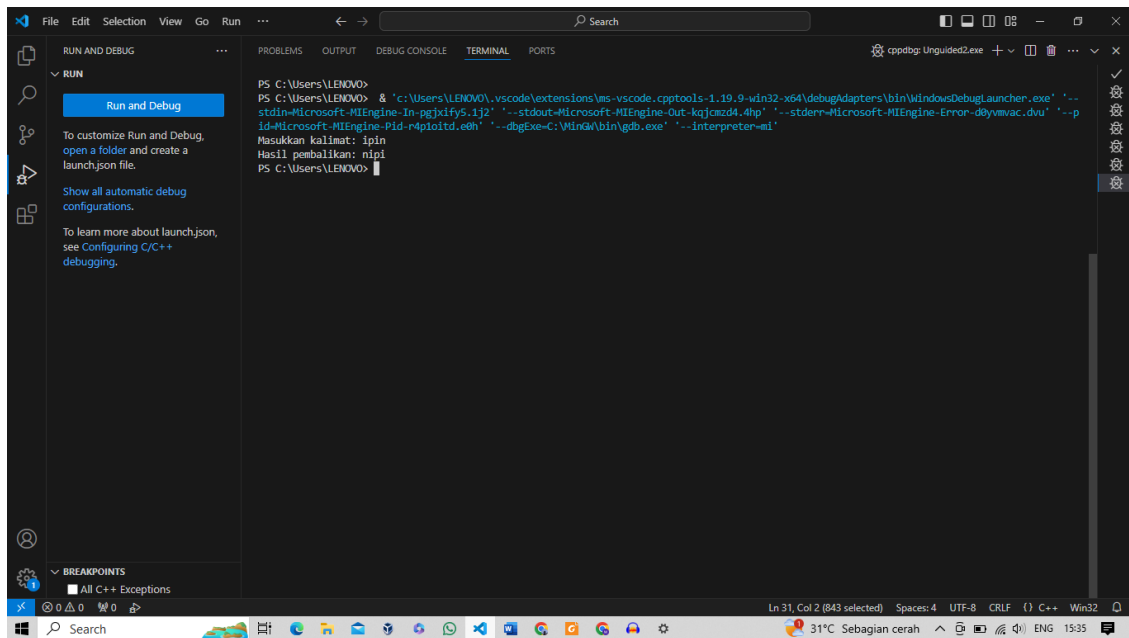
//membalikkan kalimat menggunakan stack
string reverseSentence_091(string sentence) {
    stack<char> charStack_091;
    string reversedSentence_091 = "";
    //push setiap karakter ke dalam stack
    for (char c : sentence) {
        charStack_091.push(c);
    }
    //pop setiap karakter dari stack untuk mendapatkan kalimat terbalik
    while (!charStack_091.empty()) {
        reversedSentence_091 += charStack_091.top();
        charStack_091.pop();
    }
    return reversedSentence_091;
}

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

    string hasilPembalikan = reverseSentence_091(kalimat);
    cout << "Hasil pembalikan: " << hasilPembalikan << endl;

    return 0;
}
```

### Screenshots Output :



## Deskripsi :

Program ini bertujuan fungsi `reverseSentence_091`, sebuah stack karakter (`charStack_091`) digunakan untuk menyimpan setiap karakter dari kalimat yang dimasukkan. Setelah semua karakter dimasukkan ke dalam stack, karakter-karakter tersebut kemudian diambil satu per satu dari stack dan disusun kembali dalam urutan terbalik, membentuk kalimat yang sudah dibalik.

Di dalam fungsi main, pengguna diminta untuk memasukkan sebuah kalimat. Kalimat tersebut kemudian diproses oleh fungsi `reverseSentence_091` untuk dibalikkan. Hasil pembalikan kalimat tersebut kemudian ditampilkan kepada pengguna. Pendekatan ini memanfaatkan sifat tumpukan (stack) yang menerapkan prinsip LIFO (Last In, First Out), di mana karakter-karakter yang dimasukkan terakhir akan dikeluarkan atau diambil terlebih dahulu. Dengan demikian, string hasil pembalikan akan terbentuk dengan karakter-karakter terakhir dari kalimat yang dimasukkan menjadi karakter pertama di string hasil pembalikan.

## D. Kesimpulan

Pertama, kode pertama menunjukkan implementasi dari struktur data tumpukan (stack) menggunakan array. Dalam kode tersebut, fungsi-fungsi dasar seperti `push`, `pop`, `peek`, dan lainnya digunakan untuk mengelola tumpukan data buku. Kedua, kode kedua menunjukkan cara untuk membersihkan karakter non-alfabet dari sebuah string dan menentukan apakah string tersebut merupakan palindrom atau tidak. Ini memperlihatkan penggunaan fungsi-fungsi string dan logika pemrograman untuk manipulasi string. Terakhir, kode ketiga menggambarkan bagaimana struktur data tumpukan (stack) bisa digunakan untuk membalikkan sebuah kalimat. Dengan memasukkan karakter-karakter kalimat ke dalam tumpukan dan kemudian

mengambilnya kembali dalam urutan terbalik, kita dapat mencapai pembalikan kalimat tersebut.

#### E. Referensi

Adieb, M. (2021) *Stack: Pengertian, Kelebihan dan Kekurangan serta Jenis Operasinya*.

<https://glints.com/id/lowongan/stack-adalah/>.

Studocu (no date) *Struktur Data - Stack in c++ - STACK Salah satu konsep yang efektif untuk menyimpan dan mengambil* - Studocu. <https://www.studocu.com/id/document/universitas-muhammadiyah-prof-dr-hamka/praktikum-struktur-data/struktur-data-stack-in-c/33295526>.

*nblognlife* (2014). <https://www.nblognlife.com/2014/04/stack-pada-c.html>.