

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL VI
STACK**



**Disusun Oleh :
WIDARI DWI HAYATI
2311102060**

**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 struktur data yang sederhana dan efektif untuk menyimpan data, mirip dengan Linked Lists. Urutan data yang masuk sangat berpengaruh dalam stack. Contoh nyata dari stack adalah tumpukan piring di kafetaria. Piring-piring tersebut ditambahkan ke tumpukan saat dibersihkan dan diletakkan di atas, dan ketika dibutuhkan, piring tersebut diambil dari atas tumpukan. Piring yang pertama diletakkan di tumpukan adalah yang terakhir digunakan.

Definisi: Stack adalah daftar yang terurut, di mana elemen-elemen ditambahkan dan dihapus hanya dari satu ujung, yaitu bagian atas. Elemen yang terakhir dimasukkan akan menjadi yang pertama dihapus, sehingga disebut sebagai 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.
- i. Search (Cari): Mencari keberadaan elemen tertentu dalam tumpukan.

B. Guided
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;
}

```

Screenshots Output

```

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\ADVAN>

```

Nama	NIM	Kelas
Widari Dwi Hayati	2311102060	IF-11-B

Deskripsi:

Program ini adalah implementasi dari struktur data stack menggunakan array. Fungsi-fungsi yang tersedia dalam program ini yaitu `pushArrayBuku(string data)`: Menambahkan data baru ke stack. Jika stack sudah penuh, maka akan menampilkan pesan "Data telah penuh". `popArrayBuku()`: Menghapus data teratas dari stack. Jika stack kosong, maka akan menampilkan pesan "Tidak ada data yang dihapus". `peekArrayBuku(int posisi)`: Menampilkan data pada posisi tertentu dalam stack. Jika stack kosong, maka akan menampilkan pesan "Tidak ada data yang bisa dilihat". `countStack()`: Menghitung jumlah data dalam stack. `changeArrayBuku(int posisi, string data)`: Mengubah data pada posisi tertentu dalam stack. Jika posisi melebihi jumlah data, maka akan menampilkan pesan "Posisi melebihi data yang ada". `destroyArrayBuku()`: Menghapus semua data dalam stack. `cetakArrayBuku()`: Menampilkan semua data dalam stack. Jika stack kosong, maka akan menampilkan pesan "Tidak ada data yang dicetak".

Dalam fungsi `main()`, program ini melakukan beberapa operasi pada stack yaitu Menambahkan 5 buku ke stack menggunakan `pushArrayBuku()`. Menampilkan semua data dalam stack menggunakan `cetakArrayBuku()`. Menampilkan status stack apakah penuh atau kosong menggunakan `isFull()` dan `isEmpty()`. Menampilkan data pada posisi tertentu dalam stack menggunakan `peekArrayBuku()`. Menghapus data teratas dari stack menggunakan `popArrayBuku()`. Menghitung jumlah data dalam stack menggunakan `countStack()`. Mengubah data pada posisi tertentu dalam stack menggunakan `changeArrayBuku()`. Menghapus semua data dalam stack menggunakan `destroyArrayBuku()`. Menampilkan status stack setelah dihapus menggunakan `cetakArrayBuku()`.

C. Unguided/Tugas

1. Buatlah program untuk menentukan apakah kalimat tersebut yang diinputkan dalam program stack adalah palindrom/tidak. Palindrom kalimat yang dibaca dari depan dan belakang sama. Jelaskan bagaimana cara kerja programnya.

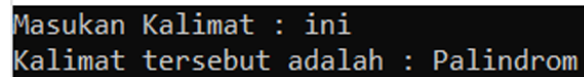
Contoh:

Kalimat : ini

Kalimat tersebut adalah polindrom

Kalimat : telkom

Kalimat tersebut adalah bukan polindrom



```
Masukan Kalimat : ini
Kalimat tersebut adalah : Palindrom
```

Source Code

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

const int maksimal = 100;
string stackKalimat[maksimal];
int top = 0;

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

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

void push (string data) {
    if (isFull()) {
        cout << "Stack telah penuh" << endl;
    } else {
        stackKalimat[top] = data;
        top++;
    }
}

void pop () {
    if (isEmpty()) {
        cout << "Tidak ada data yang dihapus" << endl;
    } else {
        top--;
    }
}
```

```

}

bool isPalindrome(string s) {
    for (int i = 0; i < s.length(); i++) {
        push(string(1, s[i]));
    }
    for (int i = 0; i < s.length(); i++) {
        if (stackKalimat[top - 1] != string(1, s[i])) {
            return false;
        }
        pop();
    }
    return true;
}

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;
}

```

Screenshots Output

The screenshot shows a Windows command prompt window with the following text:

```

Masukkan kalimat: kasur rusak
Kalimat tersebut adalah palindrom
PS C:\Users\ADVAN> ^C
PS C:\Users\ADVAN>
PS C:\Users\ADVAN> & 'c:\Users\ADVAN\.vscode\extensions\ms-vscode.cpptools-1.
Microsoft-MIEngine-In-u0x0qkn0.osa' '--stdout=Microsoft-MIEngine-Out-tbib42oe.
ft-MIEngine-Pid-5cz2oqc0.vok' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--inter
Masukkan kalimat: makan minum
Kalimat tersebut bukan palindrom
PS C:\Users\ADVAN> 

```

Overlaid on the bottom right is a VS Code window titled 'nar' with a menu bar (File, Edit, View) and a settings gear icon. The main area displays student information:

```

Nama      : Widari Dwi Hayati
NIM       : 2311102060
Kelas    : IF-11-B

```

The status bar at the bottom of the VS Code window shows: Ln 1, Col 1 | 60 characters | 100% | Window | UTF-8.

Deskripsi:

Di awal program, terdefinisi sebuah stack dengan nama `stackKalimat` dengan kapasitas maksimal maksimal dan indeks atas top awalnya 0.

Fungsi `isFull()` dan `isEmpty()` digunakan untuk mengecek apakah stack penuh atau kosong.

Fungsi `push()` digunakan untuk menambahkan data ke dalam stack. Jika stack penuh, maka akan menampilkan pesan "Stack telah penuh".

Fungsi `pop()` digunakan untuk menghapus data teratas dari stack. Jika stack kosong, maka akan menampilkan pesan "Tidak ada data yang dihapus".

Fungsi `isPalindrome()` digunakan untuk mengecek apakah suatu kalimat adalah palindrom atau bukan. Fungsi ini bekerja dengan cara mengisi stack dengan karakter-karakter dari kalimat, lalu membandingkan karakter-karakter tersebut dengan karakter-karakter yang diambil dari stack. Jika semua karakter cocok, maka kalimat tersebut adalah palindrom.

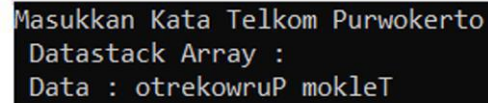
Di fungsi `main()`, program meminta pengguna untuk memasukkan kalimat. Lalu, program memanggil fungsi `isPalindrome()` untuk mengecek apakah kalimat tersebut adalah palindrom atau bukan. Jika kalimat adalah palindrom, maka program akan menampilkan pesan "Kalimat tersebut adalah palindrom". Jika bukan, maka akan menampilkan pesan "Kalimat tersebut bukan palindrom".

2. Buatlah program untuk melakukan pembalikan terhadap kalimat menggunakan stack dengan minimal 3 kata. Jelaskan output program dan source codenya beserta operasi/fungsi yang dibuat?

Contoh

Kalimat : Telkom Purwokerto

Hasil : otrekowruP mokleT



```
Masukkan Kata Telkom Purwokerto
Datastack Array :
Data : otrekowruP mokleT
```

Source Code

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

const int maksimal = 100;
string stackKalimat[maksimal];
int top = 0;

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

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

void push (string data) {
    if (isFull()) {
        cout << "Stack telah penuh" << endl;
    } else {
        stackKalimat[top] = data;
        top++;
    }
}

void pop () {
    if (isEmpty()) {
        cout << "Tidak ada data yang dihapus" << endl;
    } else {
        top--;
    }
}

string atas() {
    if (isEmpty()) {
```

```

        cout << "Stack kosong" << endl;
        return "";
    } else {
        return stackKalimat[top-1];
    }
}

string reverseLetters(string input) {
    for (int i = 0; i < input.length(); i++) {
        string temp(1, input[i]);
        push(temp);
    }

    string output = " ";
    while (!isEmpty()) {
        output += atas();
        pop();
    }

    return output;
}

string reverseWords(string input) {
    string word = "";
    for (int i = 0; i < input.length(); i++) {
        if (input[i] == ' ') {
            push(reverseLetters(word));
            word = "";
        } else {
            word += input[i];
        }
    }
    push(reverseLetters(word));

    string output = "";
    while (!isEmpty()) {
        output += atas();
        pop();
        if (!isEmpty()) {
            output += ' ';
        }
    }

    return output;
}

int main() {

```

```

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

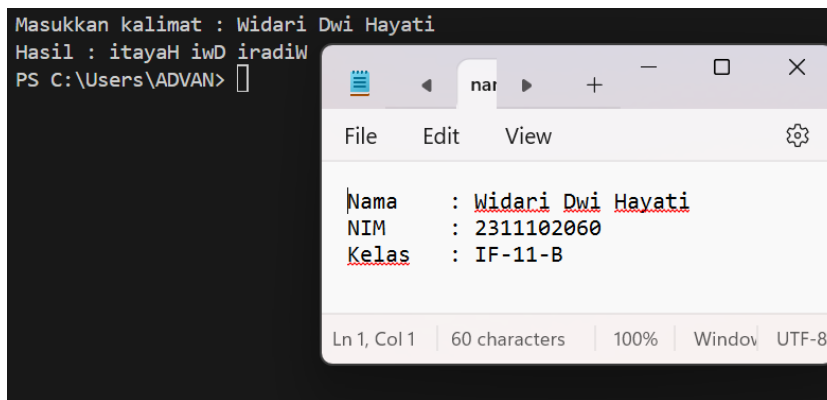
string result = reverseWords(input);

cout << "Hasil : " << result << endl;

return 0;
}

```

Screenshots Output



Deskripsi:

isFull(), isEmpty(): Fungsi-fungsi ini memeriksa apakah stack penuh (semua elemen digunakan) atau kosong (tidak ada elemen), masing-masing. Mereka mengembalikan true jika kondisi terpenuhi, dan false sebaliknya.

push(string data): Fungsi ini menambahkan string (data) ke bagian atas stack. Pertama-tama ia memeriksa apakah stack penuh. Jika tidak, ia mendorong data ke stack dan meningkatkan indeks top.

pop(): Fungsi ini menghapus elemen dari atas stack. Pertama-tama ia memeriksa apakah stack kosong. Jika tidak, ia mengurangi indeks top.

atas(): Fungsi ini mengembalikan elemen di atas stack tanpa menghapusnya. Pertama-tama ia memeriksa apakah stack kosong. Jika tidak, ia mengembalikan elemen pada indeks top-1.

reverseLetters(string input): Fungsi ini mengambil string sebagai input dan mengembalikan string baru dengan karakter dibalik. Ini mendorong setiap karakter string input ke stack, kemudian melepasnya satu per satu dan menambahkannya ke string output.

`reverseWords(string input)`: Fungsi ini mengambil kalimat (string) sebagai input dan mengembalikan string baru dengan kata-kata dibalik sambil mempertahankan spasi. Ini mengulangi string input:

Jika menemukan spasi, ia membalikkan kata saat ini menggunakan `reverseLetters` dan mendorongnya ke stack. Ini kemudian mengatur ulang string word untuk kata berikutnya. Jika tidak, ia membangun kata saat ini dengan menambahkan karakter. Setelah memproses seluruh input, ia membalikkan kata terakhir dan mendorongnya ke stack. Akhirnya, mengulangi stack, mengeluarkan setiap kata, menambahkannya ke string output, dan menambahkan spasi di antara kata (kecuali setelah kata terakhir).

Seperti pada output ketika user menginputkan “Widari Dwi Hayati”, program akan mengouputkan kalimat yang diinputkan secara terbalik yaitu “itayaH iwD iradiW”.

D. Kesimpulan

Stack adalah daftar terurut di mana elemen ditambahkan dan dihapus hanya dari satu ujung, yaitu bagian atas. Elemen terakhir yang dimasukkan akan menjadi yang pertama dihapus, sehingga disebut sebagai LIFO (Last In First Out).

Operasi pada Stack:

Push (Masukkan): Menambahkan elemen ke stack.

Pop (Keluarkan): Menghapus elemen dari stack.

Top (Atas): Mendapatkan nilai elemen teratas.

IsEmpty (Kosong): Memeriksa apakah stack kosong.

IsFull (Penuh): Memeriksa apakah stack penuh.

Size (Ukuran): Mengembalikan jumlah elemen dalam stack.

Peek (Lihat): Melihat nilai elemen pada posisi tertentu.

Clear (Hapus Semua): Mengosongkan stack.

Search (Cari): Mencari elemen tertentu dalam stack.

E. Referensi (APA)

Stroustrup, B. (2021). A Tour of C++ (2nd ed.). Addison-Wesley Professional.

Lippman, S. L., Lajoie, J., & Moo, B. (2020). C++ Primer (6th ed.). Addison-Wesley Professional.

Malik, D. S. (2014). C++ Programming: From Problem Analysis to Program Design (7th ed.). Cengage Learning.