

LAPORAN PRAKTIKUM

MODUL VI STACK



Disusun oleh:
Muhammad Agha Zulfadhli
NIM: 2311102015

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

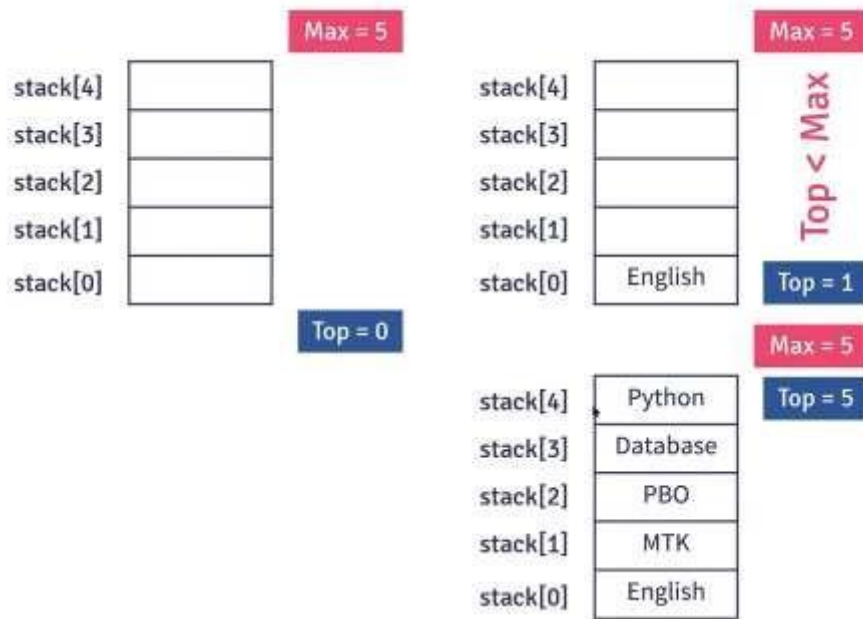
- a. Mampu memahami konsep stack pada struktur data dan algoritma
- b. Mampu mengimplementasikan operasi-operasi pada stack
- c. Mampu memecahkan permasalahan dengan solusi stack

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:

- Push (Masukkan):** Menambahkan elemen ke dalam tumpukan pada posisi paling atas atau ujung.
- 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.

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;
    }
}
```

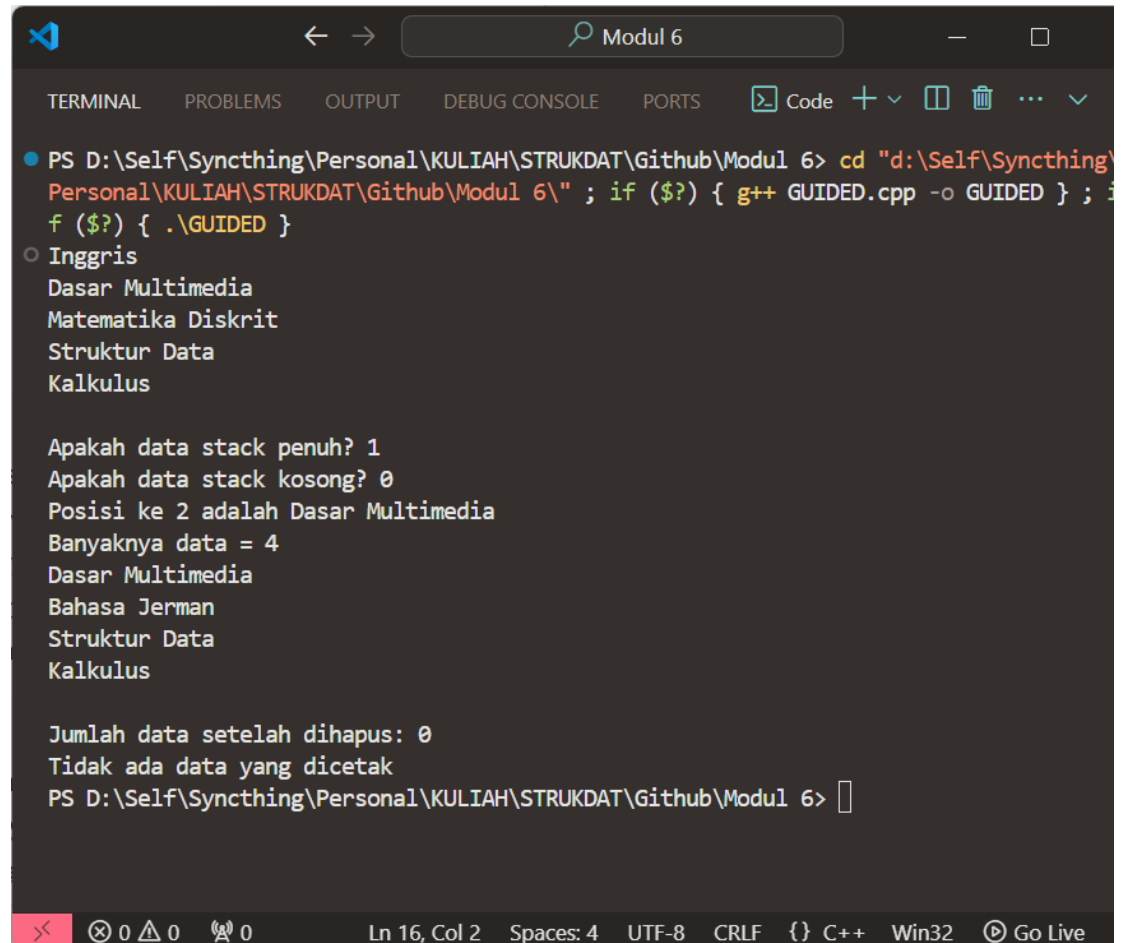
```

    }
    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 D:\Self\Syncthing\Personal\KULIAH\STRUKDAT\Github\Modul 6> cd "d:\Self\Syncthing\Personal\KULIAH\STRUKDAT\Github\Modul 6\" ; if ($?) { g++ GUIDED.cpp -o GUIDED } ; if ($?) { .\GUIDED }
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 D:\Self\Syncthing\Personal\KULIAH\STRUKDAT\Github\Modul 6> 
```

Deskripsi program

Program di atas adalah sebuah implementasi dari struktur data stack menggunakan array. Program ini memiliki beberapa fungsi untuk memanipulasi dan mengelola data dalam array yang bertindak sebagai stack. Array `arrayBuku` digunakan untuk menyimpan data buku, dengan ukuran maksimal maksimal. Variabel `top` menunjukkan indeks dari elemen teratas dalam stack.

Fungsi `isFull()` dan `isEmpty()` digunakan untuk memeriksa apakah `stack` penuh atau kosong. Fungsi `pushArrayBuku()` menambahkan elemen baru ke dalam `stack` jika `stack` belum penuh. Fungsi `popArrayBuku()` menghapus elemen teratas dari `stack` jika `stack` tidak kosong. Fungsi `peekArrayBuku()` digunakan untuk melihat elemen pada posisi tertentu dalam `stack`. Fungsi `countStack()` mengembalikan jumlah elemen dalam `stack`. Fungsi `changeArrayBuku()` mengganti data pada posisi tertentu dalam `stack`. Fungsi `destroyArrayBuku()` menghapus semua data dalam `stack`.

Di dalam `main()` function, beberapa operasi dilakukan pada `stack` buku, seperti menambahkan, menghapus, melihat, dan mengubah data. Hasil operasi tersebut kemudian dicetak untuk ditampilkan kepada pengguna. Program juga menampilkan apakah `stack` penuh atau kosong setelah beberapa operasi dilakukan. Akhirnya, `stack` dihapus dan statusnya dicetak kembali untuk menunjukkan bahwa `stack` telah kosong.

LATIHAN KELAS - UNGUIDED

Unguided 1

Source code

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

void printStack(char *stack, const int &size) {
    for (int i = 0; i < size; i++) cout << stack[i];
    cout << endl;
}

void swapStack(char *stack, const int &size) {
    for (int i = 0, j = size - 1; i < j; i++, j--) {
        swap(stack[i], stack[j]);
    }
}

bool checkPolindrom(const char *stack_ori, const char
*stack_swapped, const int &size) {
    for (int i = 0; i < size; i++) {
        if (stack_ori[i] != stack_swapped[i]) return false;
    }
    return true;
}

int main()
{
    const int MAX_SIZE = 100;

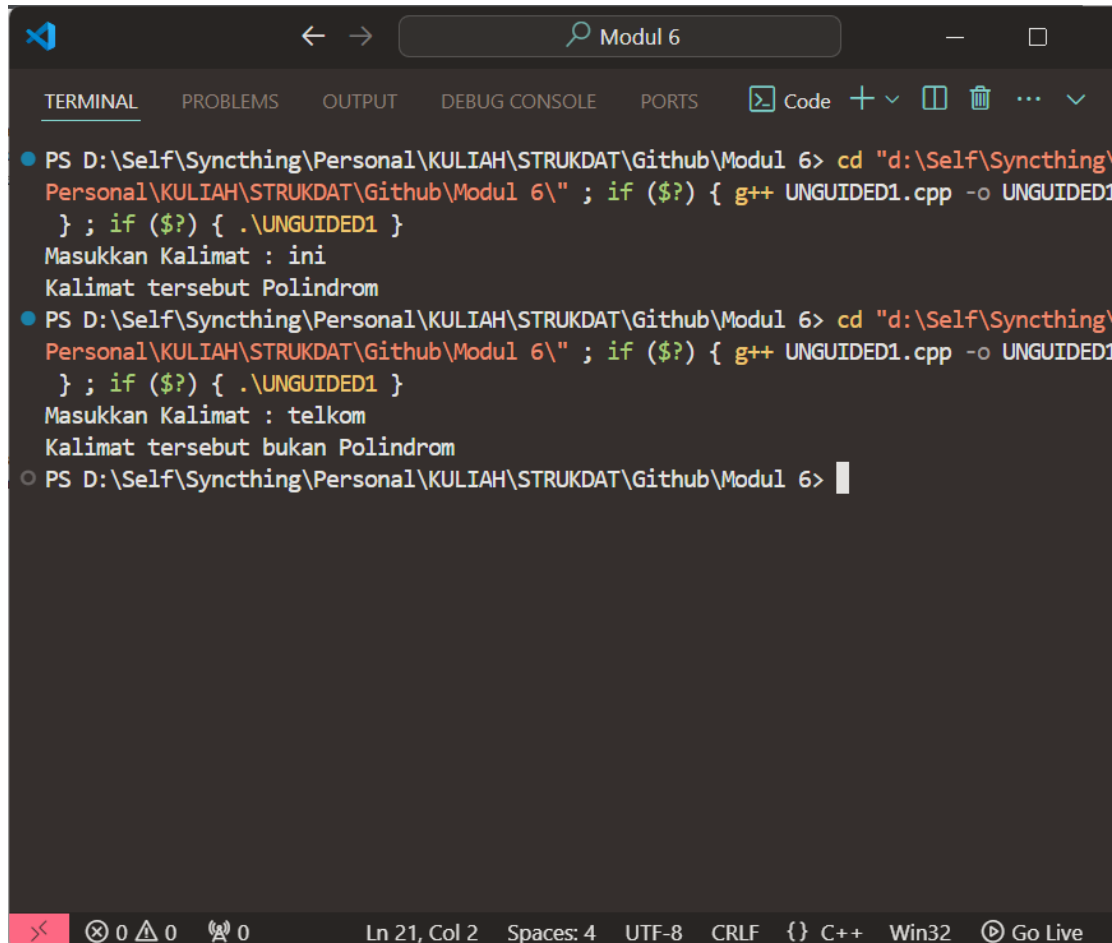
    char stack[MAX_SIZE];
    cout << "Masukkan Kalimat : ";
    cin.getline(stack, MAX_SIZE);
    // inputnya satu aja yah 🙄

    int size = strlen(stack);
    char stack_ori[MAX_SIZE + 1];
    strcpy(stack_ori, stack);
    swapStack(stack, size);

    if (checkPolindrom(stack_ori, stack, size)) cout << "Kalimat
tersebut Polindrom" << endl;
    else cout << "Kalimat tersebut bukan Polindrom" << endl;

    return 0;
}
```

Screenshoot program



```
Modul 6

TERMINAL  PROBLEMS  OUTPUT  DEBUG CONSOLE  PORTS  Code  +  -  ...  v

• PS D:\Self\Syncthing\Personal\KULIAH\STRUKDAT\Github\Modul 6> cd "d:\Self\Syncthing\
Personal\KULIAH\STRUKDAT\Github\Modul 6\" ; if ($?) { g++ UNGUIDED1.cpp -o UNGUIDED1
} ; if ($?) { .\UNGUIDED1 }
Masukkan Kalimat : ini
Kalimat tersebut Polindrom
• PS D:\Self\Syncthing\Personal\KULIAH\STRUKDAT\Github\Modul 6> cd "d:\Self\Syncthing\
Personal\KULIAH\STRUKDAT\Github\Modul 6\" ; if ($?) { g++ UNGUIDED1.cpp -o UNGUIDED1
} ; if ($?) { .\UNGUIDED1 }
Masukkan Kalimat : telkom
Kalimat tersebut bukan Polindrom
○ PS D:\Self\Syncthing\Personal\KULIAH\STRUKDAT\Github\Modul 6> |

><  0 0 0  Ln 21, Col 2  Spaces: 4  UTF-8  CRLF  {} C++  Win32  Go Live
```

Deskripsi program

Program di atas adalah sebuah program yang bertujuan untuk mengecek apakah sebuah kalimat yang dimasukkan oleh pengguna merupakan palindrom atau bukan. Palindrom adalah sebuah kata, frasa, angka, atau susunan lainnya yang dapat dibaca dengan sama baik dari depan maupun dari belakang. Program ini menggunakan pendekatan dengan menggunakan stack untuk membalikkan kalimat yang dimasukkan, kemudian membandingkan kalimat asli dengan kalimat yang sudah dibalik.

Pertama-tama, program meminta pengguna untuk memasukkan sebuah kalimat. Kalimat tersebut disimpan dalam array karakter stack dengan batas maksimum ukuran MAX_SIZE. Kemudian, program menghitung panjang kalimat yang dimasukkan menggunakan fungsi `strlen()` dari library `<cstring>`, dan menyimpan kalimat asli dalam array karakter `stack_ori` menggunakan fungsi `strcpy()`.

Setelah itu, program memanggil fungsi `swapStack()` untuk membalikkan urutan karakter dalam kalimat yang dimasukkan. Fungsi ini melakukan pertukaran karakter dari indeks pertama dengan indeks terakhir, kedua dengan kedua dari ujung, dan seterusnya hingga mencapai tengah dari kalimat.

Selanjutnya, program membandingkan kalimat asli dengan kalimat yang sudah dibalik menggunakan fungsi `checkPolindrom()`. Fungsi ini membandingkan setiap karakter dari kedua kalimat, jika terdapat perbedaan pada salah satu karakter, maka program akan mengeluarkan output bahwa kalimat tersebut bukan palindrom. Jika tidak ada perbedaan, maka kalimat tersebut akan dianggap sebagai palindrom.

Terakhir, program mencetak hasil pengecekan, apakah kalimat yang dimasukkan merupakan palindrom atau bukan, berdasarkan hasil evaluasi dari fungsi `checkPolindrom()`.

Unguided 2

Source code

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

void printStack(char *stack, const int &size) {
    for (int i = 0; i < size; i++) cout << stack[i];
    cout << endl;
}

void swapStack(char *stack, const int &size) {
    for (int i = 0, j = size - 1; i < j; i++, j--) {
        swap(stack[i], stack[j]);
    }
}

int main()
{
    const int MAX_SIZE = 100;

    char stack[MAX_SIZE];
    cout << left << setw(18) << "Masukkan Kalimat " << ": ";
    cin.getline(stack, MAX_SIZE);
    // inputnya satu aja yah 🗨️

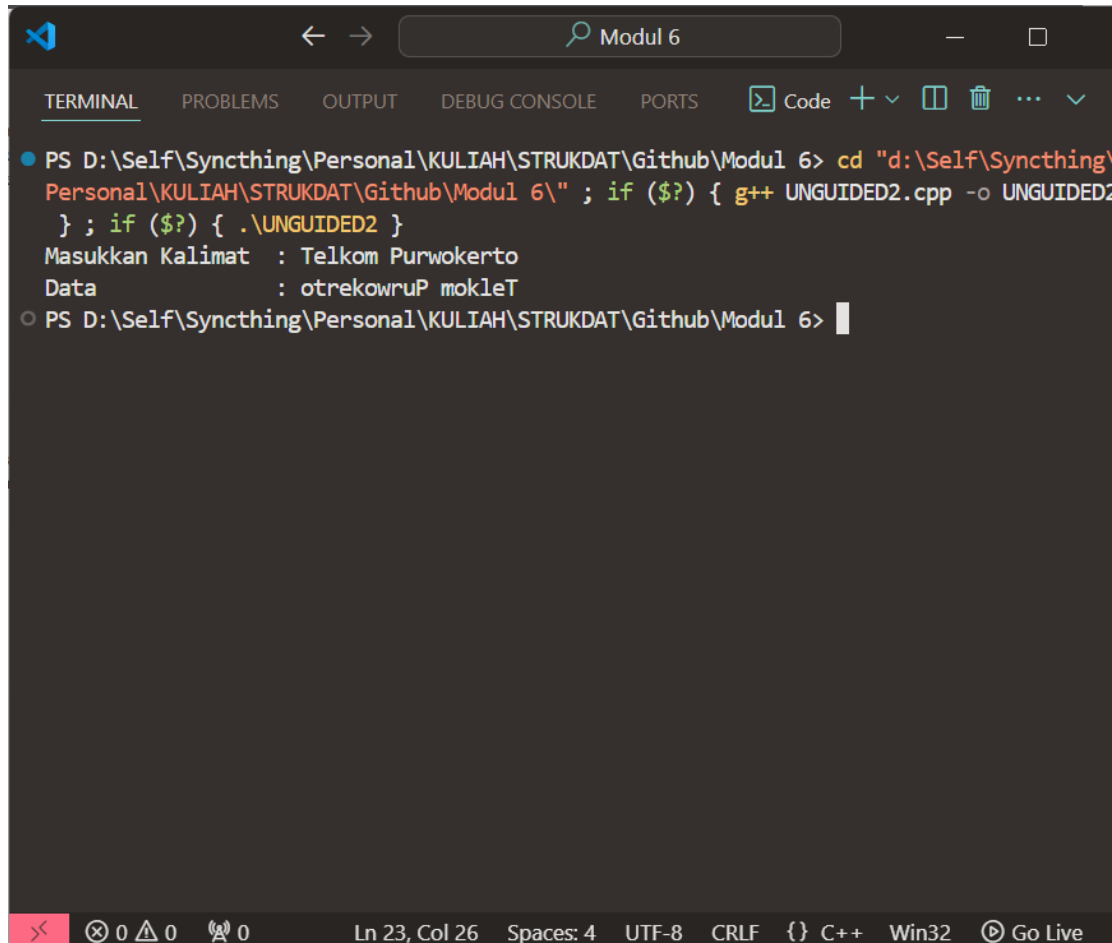
    int size = strlen(stack);

    char stack_ori[MAX_SIZE + 1];
    strcpy(stack_ori, stack);

    swapStack(stack, size);
    cout << left << setw(18) << "Data " << ": " << stack;

    return 0;
}
```

Screenshoot program



The screenshot shows a Visual Studio Code interface with a terminal window open. The terminal displays the following commands and output:

```
PS D:\Self\Syncthing\Personal\KULIAH\STRUKDAT\Github\Modul 6> cd "d:\Self\Syncthing\Personal\KULIAH\STRUKDAT\Github\Modul 6\" ; if ($?) { g++ UNGUIDED2.cpp -o UNGUIDED2 } ; if ($?) { .\UNGUIDED2 }
Masukkan Kalimat : Telkom Purwokerto
Data : otrekowruP mokleT
PS D:\Self\Syncthing\Personal\KULIAH\STRUKDAT\Github\Modul 6>
```

The status bar at the bottom indicates the file is at line 23, column 26, with 4 spaces, UTF-8 encoding, CRLF line endings, and C++ language. The window title is 'Modul 6'.

Deskripsi program

Program C++ di atas adalah sebuah program sederhana yang bertujuan untuk membalikkan sebuah kalimat yang dimasukkan oleh pengguna menggunakan pendekatan dengan stack. Program ini menggunakan tiga fungsi, yaitu `printStack`, `swapStack`, dan `main`.

Fungsi `printStack` bertujuan untuk mencetak isi dari array karakter `stack` ke dalam layar. Fungsi ini menerima dua parameter, yaitu array karakter `stack` dan ukuran `size`, dan menggunakan loop untuk mencetak setiap karakter dari `stack`.

Fungsi `swapStack` digunakan untuk membalikkan urutan karakter dalam array `stack`. Fungsi ini menerima dua parameter, yaitu array karakter `stack` dan ukuran `size`. Dalam fungsi ini, terdapat loop yang melakukan pertukaran karakter dari indeks pertama dengan indeks terakhir, kedua dengan kedua dari ujung, dan seterusnya hingga mencapai tengah dari kalimat.

Di dalam `main` function, program pertama-tama mendefinisikan konstanta `MAX_SIZE` yang menentukan batas maksimum ukuran kalimat yang dapat dimasukkan. Kemudian, program meminta pengguna untuk memasukkan sebuah kalimat menggunakan `cin.getline()` dan menyimpannya dalam array karakter `stack`. Ukuran kalimat yang dimasukkan kemudian dihitung menggunakan fungsi `strlen()` dari library `<cstring>`.

Selanjutnya, program menyimpan kalimat asli dalam array karakter `stack_ori` menggunakan fungsi `strcpy()`. Setelah itu, program memanggil fungsi `swapStack()` untuk membalikkan urutan karakter dalam kalimat yang dimasukkan. Hasil dari pembalikan tersebut kemudian dicetak ke layar menggunakan `cout`.

Program menggunakan manipulator `setw()` dari library `<iomanip>` untuk mengatur lebar output agar lebih rapi, serta `left` untuk penyesuaian ke kiri. Ini memastikan bahwa prompt dan data yang dicetak memiliki format yang baik.

BAB IV

KESIMPULAN

Dari uraian tersebut, dapat disimpulkan beberapa poin terkait Stack

1. Stack adalah struktur data sederhana yang mengikuti aturan Last In First Out (LIFO), yang berarti elemen terakhir yang dimasukkan akan menjadi yang pertama dikeluarkan. Contoh umum dari stack adalah tumpukan piring di kafetaria, di mana piring yang terakhir ditempatkan akan menjadi yang pertama diambil.
2. Operasi dasar pada stack meliputi Push (menambahkan elemen), Pop (menghapus elemen), Top (mendapatkan nilai elemen teratas tanpa menghapusnya), IsEmpty (memeriksa apakah stack kosong), IsFull (memeriksa apakah stack penuh), Size (mengembalikan jumlah elemen dalam stack), Peek (melihat nilai pada posisi tertentu tanpa menghapusnya), Clear (mengosongkan semua elemen), dan Search (mencari keberadaan elemen tertentu dalam stack).
3. Penggunaan stack umumnya terdapat dalam pemrosesan data terstruktur, algoritma rekursif, pengelolaan memori pada sistem komputer, dan implementasi struktur data seperti undo/redo functionality dalam aplikasi.

Dengan demikian, Stack merupakan struktur data yang sangat berguna dalam menyimpan dan mengelola data.

DAFTAR PUSTAKA

[1] Learning Management System ,MODUL 6 STACK.

[2] muh—agha—zul, 「PRAK」 Stack, 2024.

<https://www.youtube.com/watch?v=FcEeS88dr6Q>