

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL VI
STACK**



Disusun oleh:
Alfin Ilham Berlianto
2311102047

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

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

MODUL 6

STACK

A. Tujuan

1. Mampu memahami konsep stack pada struktur data dan algoritma.
2. Mampu mengimplementasikan operasi-operasi pada stack.
3. Mampu memecahkan permasalahan dengan sokusi stack.

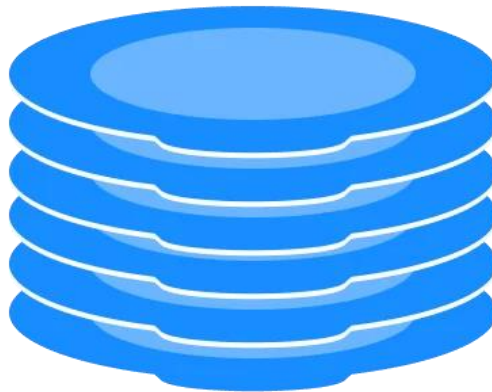
B. Dasar Teori

Struktur data merupakan aspek yang cukup penting diketahui oleh setiap orang yang memulai bidang pemrograman dan komputer. Dengan memahami prinsip dan cara kerja sebuah struktur data kita dapat mengimplementasikan ke dalam program dengan lebih baik dan sesuai fungsinya.

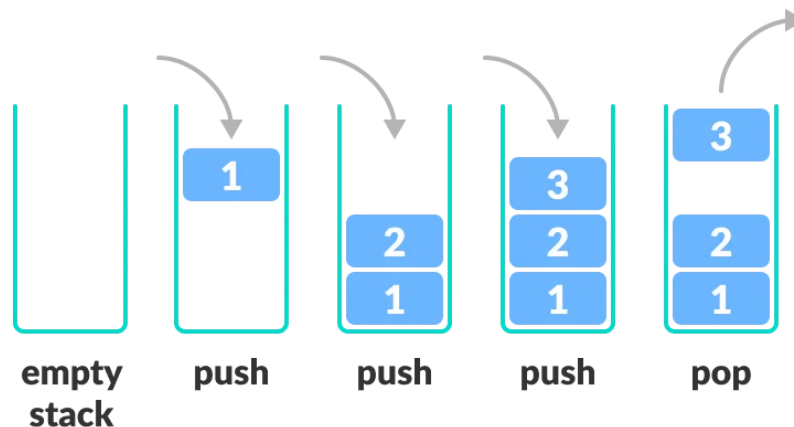
Salah satunya adalah struktur data stack. Apa itu stack? Bagaimana cara kerjanya?

Stack atau dalam Bahasa Indonesia diartikan tumpukan, adalah struktur data linier yang mengikuti prinsip Last In First Out (LIFO). Artinya elemen yang terakhir disisipkan akan menjadi elemen pertama yang keluar.

Cara struktur data stack dalam menyimpan sebuah nilai dapat kita bayangkan seperti piring yang disusun rapi secara bertumpuk ke atas. Apabila kita ingin mengambil piring bagian bawah, kita harus terlebih dahulu menyisihkan semua piring yang ada di atas.



Dalam istilah pemrograman, upaya menambahkan elemen pada struktur data stack disebut dengan push. Sedangkan proses menghapus atau menghilangkan elemen data dari stack disebut pop.



Dari gambar di atas, dapat terlihat bahwa meskipun elemen ke-3 adalah yang paling terakhir ditambahkan, namun elemen tersebut justru yang pertama dihapus. Operasi inilah yang kemudian disebut sebagai prinsip operasi LIFO (Last In First Out).

Kita dapat mengimplementasikan stack dengan bahasa pemrograman seperti C, C++, Java, Python, atau C#.

Perlu diketahui, dalam struktur data stack ada dua kondisi yang perlu dihindari, yaitu underflow dan overflow.

- Stack underflow, yaitu keadaan dimana kita mencoba mengakses atau menghapus elemen data pada stack yang kosong.
- Stack overflow, yaitu keadaan di mana ruang memori yang dialokasikan untuk struktur data stack sudah penuh namun masih dilakukan operasi penyisipan elemen

Jenis Jenis Stack

Berdasarkan kemampuan menyimpan data, struktur data stack dapat dibagi menjadi 2 jenis, yaitu: register stack dan memory stack.

1. Register stack

Register stack merupakan stack yang hanya mampu menampung data dalam jumlah yang kecil. Kedalaman maksimum pada register stack cenderung dibatasi karena ukuran unit memorinya sangat kecil dibandingkan dengan memory stack.

2. Memory stack

Pada stack jenis ini, kedalaman dari stack cukup fleksibel dan mampu menangani dalam dalam skala yang lebih besar dibandingkan jenis sebelumnya.

Operasi-operasi dasar pada stack

Ada beberapa operasi dasar yang bisa kita lakukan terhadap struktur data stack. Operasi-operasi tersebut meliputi

- Push: Menyisipkan elemen ke bagian atas stack
- Pop: Menghapus elemen atas dari stack
- IsEmpty: Memeriksa apakah stack kosong
- IsFull: Memeriksa apakah stack sudah penuh
- Peek: Mendapatkan nilai elemen teratas tanpa menghapusnya

Fungsi dan Kegunaan Stack

Adapun fungsi dan kegunaan struktur data stack adalah sebagai berikut:

- Struktur data stack digunakan dalam evaluasi dan konversi ekspresi aritmatika. Proses ini banyak dipakai untuk program kompiler.
- Stack digunakan dalam pemrograman rekursi.
- Digunakan untuk pemeriksaan tanda kurung.
- Stack digunakan dalam manajemen memori.
- Dipakai untuk memproses pemanggilan sebuah fungsi.

Salah satu contoh penerapan struktur data stack adalah fitur tombol back pada browser. Dimana browser akan menyimpan semua URL yang telah kita kunjungi sebelumnya dalam stack.

Setiap kali kita mengunjungi halaman baru, halaman itu ditambahkan di atas stack. Saat kita menekan tombol kembali, URL saat ini dihapus dari tumpukan, dan URL sebelumnya diakses.

C. 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;
}

```

SCREENSHOT OUTPUT

```

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

PS C:\Users\sjsjs\Desktop\code c++\Guided 6 1> cd "c:\Users\sjsjs\Desktop\code c++\Guided 6 1\" ; 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\sjsjs\Desktop\code c++\Guided 6 1>

```

DESKRIPSI PROGRAM

Pada guided ini membuat program dengan stack, dalam sub-program terdapat fungsi dan prosedur, diantaranya adalah bool isFull() yang berfungsi untuk mengetahui kondisi apakah stacknya sudah penuh atau belum, bool isEmpty() untuk mengetahui apakah stacknya masih kosong atau sudah terisi, void pushArrayBuku() untuk menambahkan data ke dalam stack, void popArrayBuku() untuk mengambil/menghapus data yang ada di dalam stack, void peekArrayBuku() untuk melihat satu-satu data di dalam stack tanpa harus menghapusnya, int countStack() untuk mengetahui jumlah data yang ada di dalam stack, void changeArrayBuku() untuk mengedit data di dalam stack, void destroyArrayBuku() untuk mengkosongkan data yang ada di dalam stack, void cetakArrayBuku() untuk menampilkan output dari stack yang sudah dibuat. Lalu di dalam main program untuk mengisi void pushArrayBuku() dengan menggunakan parameter string data, dan diisi statis/manual dengan inputan yang tertera di screenshot, lalu menampilkan apakah stack itu sudah penuh atau belum, apakah stack itu kosong atau tidak, hasilnya stack itu menghasilkan 1(true) artinya sudah penuh, dan apakah stack itu kosong atau tidak menghasilkan 0(false) artinya stack itu tidak kosong.

D. Unguided

Unguided 1

SOURCE CODE

```
#include <iostream>
#include <stack>
#include <string>

using namespace std;

bool isPalindrome(const string &word) {
    stack<char> charStack;

    for (char ch : word) {
        charStack.push(ch);
    }

    string reversedWord;
    while (!charStack.empty()) {
        reversedWord += charStack.top();
        charStack.pop();
    }

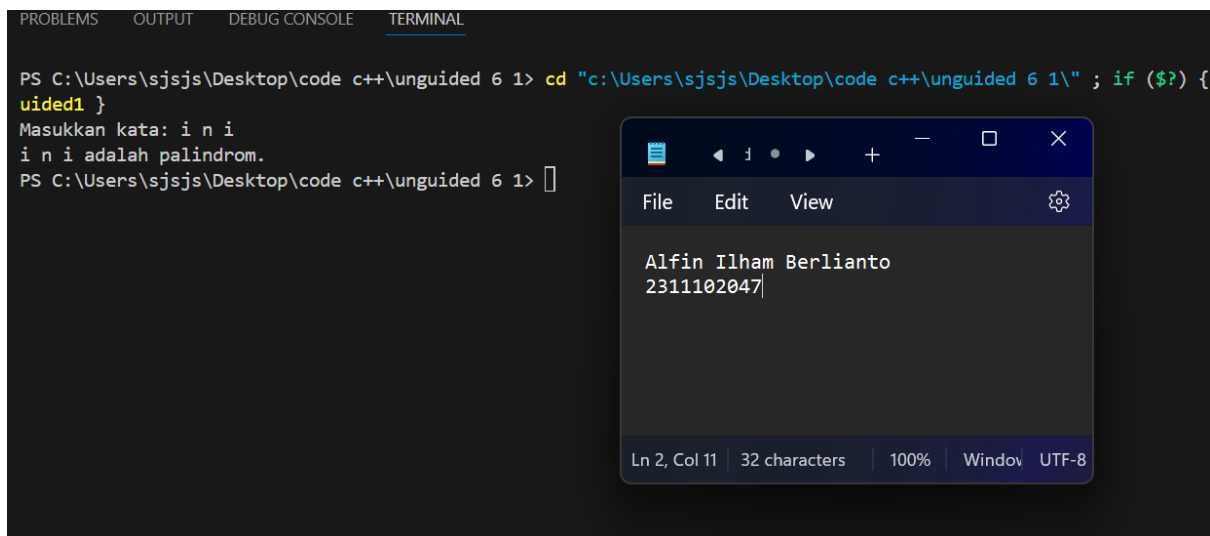
    return word == reversedWord;
}

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

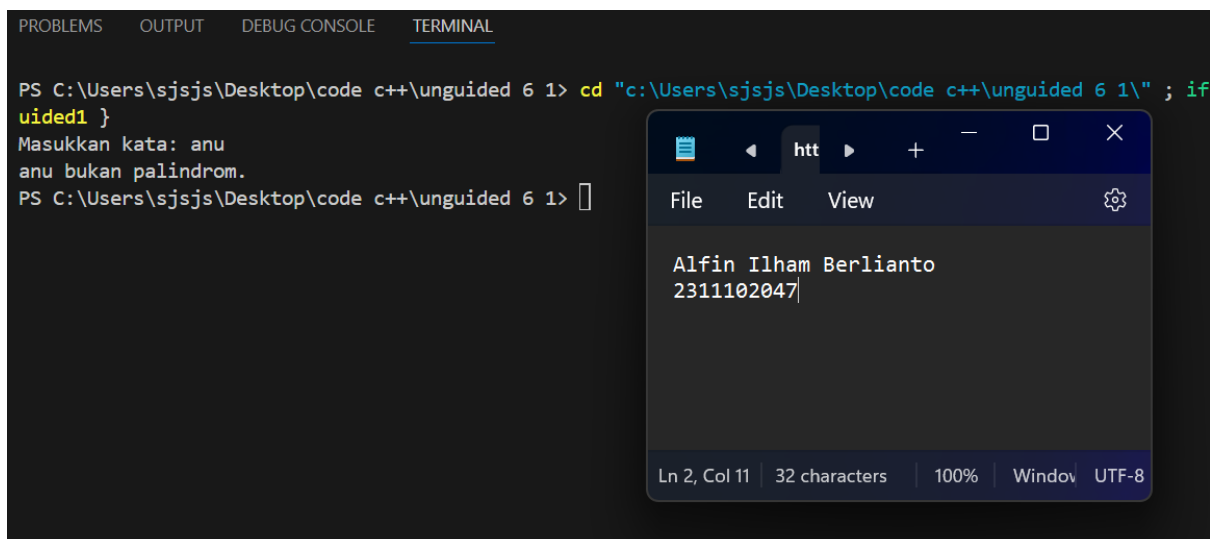
    if (isPalindrome(word)) {
        cout << word << " adalah palindrom." << endl;
    } else {
        cout << word << " bukan palindrom." << endl;
    }

    return 0;
}
```

SCREENSHOT OUTPUT



```
PS C:\Users\sjsjs\Desktop\code c++\unguided 6 1> cd "c:\Users\sjsjs\Desktop\code c++\unguided 6 1\" ; if ($?) {
    uided1 }
Masukkan kata: i n i
i n i adalah palindrom.
PS C:\Users\sjsjs\Desktop\code c++\unguided 6 1> 
```



```
PS C:\Users\sjsjs\Desktop\code c++\unguided 6 1> cd "c:\Users\sjsjs\Desktop\code c++\unguided 6 1\" ; if
    uided1 }
Masukkan kata: anu
anu bukan palindrom.
PS C:\Users\sjsjs\Desktop\code c++\unguided 6 1> 
```

DESKRIPSI PROGRAM

Pada unguided ini mmebuat program mencari kata palindrom atau bukan menggunakan stack, cara pembuatannya dalam sub program membuat fungsi Boolean untuk menampilkan apakah kata yang diinputkan itu palindrom atau bukan, yaitu di dalam fungsi Boolean terdapat perulangan for untuk memasukan data char ch dari variabel word, lalu selama variabel charstack nya tidak kosong variabel reverseWord akan mengambil satu-satu nilai atas dari kata yang dari variabel word,lalu di returnkan dengan perkondisian ternary operator,apakah nilai variabel word dengan nilai variabel reverseWord itu sama? Jika sama bernilai true dan jika tidak bernilai false.

Lalu pada main programnya terdapat prompt untuk mengisi user ingin memilih kata apa,jika sudah maka akan masuk ke perkondisian, jika kata yang diinputkan bernilai true dari fungsi Boolean yang sudah dieksekusi,maka akan menampilkan kata yang diinputkan adalah kata palindrom,jika bernilai false maka kata yang diinputkan bukan kata palindrom.

Unguided 2

SOURCE CODE

```
#include <iostream>
#include <stack>
#include <string>

using namespace std;

string reverseWords(const string &str) {
    stack<char> s;
    string result = "";

    // Membalikkan setiap karakter dalam string menggunakan stack
    for (char ch : str) {
        s.push(ch);
    }

    // Mengambil karakter dari stack untuk membentuk kalimat yang dibalik
    while (!s.empty()) {
        result += s.top();
        s.pop();
    }

    return result;
}

int main() {
    string input;

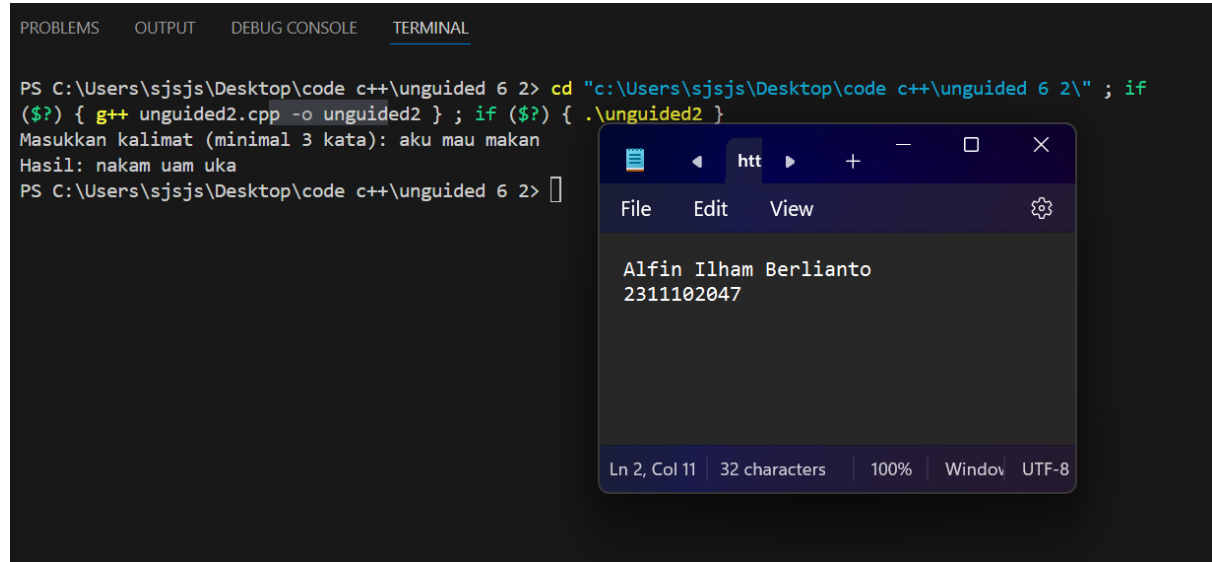
    cout << "Masukkan kalimat (minimal 3 kata): ";
    getline(cin, input);

    // Memastikan bahwa input memiliki minimal 3 kata
    int wordCount = 0;
    for (char ch : input) {
        if (ch == ' ') {
            wordCount++;
        }
    }
    wordCount++; // Menambah satu untuk kata terakhir

    if (wordCount < 3) {
        cout << "Error: Kalimat harus memiliki minimal 3 kata." << endl;
    } else {
        string reversed = reverseWords(input);
        cout << "Hasil: " << reversed << endl;
    }
}
```

```
return 0;
}
```

SCREENSHOT OUTPUT



```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL
PS C:\Users\sjsjs\Desktop\code c++\unguided 6 2> cd "c:\Users\sjsjs\Desktop\code c++\unguided 6 2\" ; if ($?) { g++ unguided2.cpp -o unguided2 } ; if ($?) { .\unguided2 }
Masukkan kalimat (minimal 3 kata): aku mau makan
Hasil: nakam uam uka
PS C:\Users\sjsjs\Desktop\code c++\unguided 6 2> 
```

DESKRIPSI PROGRAM

Program ini bertujuan untuk membalikkan urutan karakter dalam sebuah kalimat yang diinputkan oleh user, dengan syarat kalimat tersebut harus memiliki minimal 3 kata. Program ini menggunakan stack untuk membalikkan urutan karakter dalam kalimat.

- Stack: digunakan untuk mengambil input kalimat dari pengguna
- Getline: digunakan untuk mengambil input kalimat dari pengguna
- Pop: menghapus karakter dari stack dan menambahkannya ke string hasil.

Lalu pada main program terdapat variabel wordcount, itu berfungsi pada perkondisian jika $ch = \text{spasi}$, artinya jika ada spasi 1 kali maka wordcount akan bertambah, itu juga untuk mengkondisikan jika spasinya kurang dari 3 kali maka akan menampilkan pesan error, jika 3 kali atau lebih maka output nya akan dieksekusi.

D. Kesimpulan

Stack adalah salah satu struktur data linear yang mengikuti prinsip LIFO (Last In, First Out), dimana elemen yang terakhir dimasukkan adalah yang pertama dikeluarkan. Di C++, stack dapat diimplementasikan menggunakan array, linked list, atau dengan memanfaatkan pustaka standar C++ (STL) seperti `std::stack`.

Poin-poin Penting tentang Stack pada C++:

Operasi Dasar:

- Push: Menambahkan elemen ke puncak stack.
- Pop: Menghapus elemen dari puncak stack.
- Top: Mengambil nilai elemen di puncak tanpa menghapusnya.
- isEmpty: Mengecek apakah stack kosong.
- Size: Mengembalikan jumlah elemen dalam stack.

DAFTAR PUSTAKA

1. Trivusi, (2022, 16 September) Struktur Data Stack: Pengertian, Karakteristik, dan Kegunaannya. Diakses pada 19 Mei 2024, dari <https://www.trivusi.web.id/2022/07/struktur-data-stack.html>
- 2, nblognlife, (2014, 26 April) Stack pada C++ Diakses pada 19 Mei 2024, dari <https://www.nblognlife.com/2014/04/stack-pada-c.html>