

**LAPORAN PRAKTIKUM
STRUKTUR DATA PEMOGRAMAN**

**MODUL VI
“STACK”**



Disusun Oleh:

NAMA : Trie Nabilla Farhah

NIM : 2311102071

Dosen Pengampu:

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

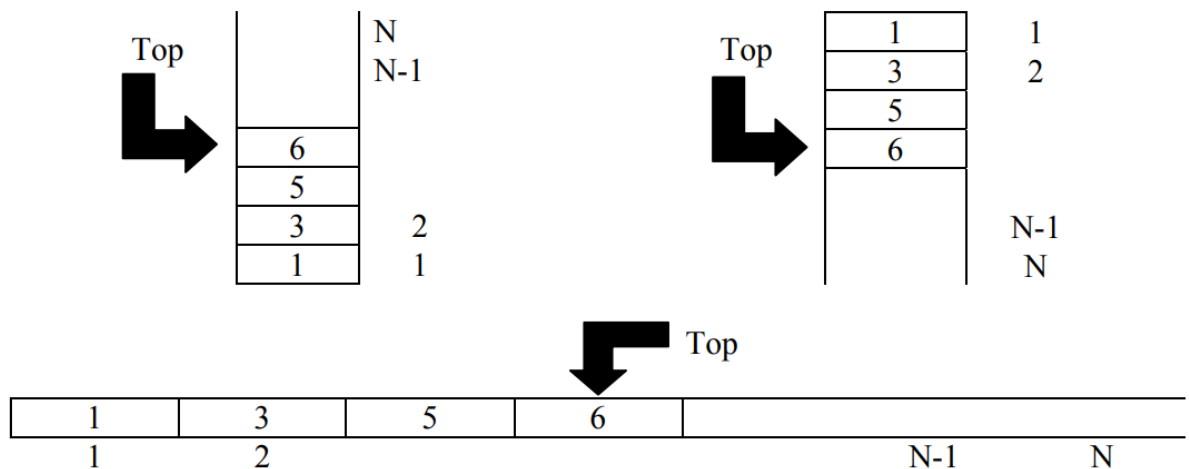
**PROGRAM STUDI S1 TEKNIK INFORMATIKA
FAKULTAS INFORMATIKA
INSTITUT TEKNOLOGI TELKOM PURWOKERTO
PURWOKERTO**

2024

A. DASAR TEORI

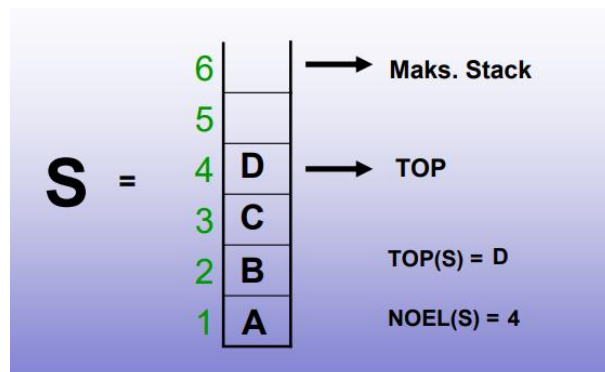
STACK adalah salah satu list linear dalam struktur data yang digunakan untuk Menyimpan dan mengambil data dengan konsep LIFO (LastInFirstOut). Dimana dalam Stack ini kumpulan data yang masuk diletakkan diatas data yang lain. Dan berdasar konsep LIFO maka data yang terakhir kali disimpan dalam stack akan menjadi data yang pertama kali diambil. Dalam prosesnya, untuk memasukkan sebuah data ke dalam stack atau dengan kata lain ke bagian atas dari sebuah tumpukan digunakan perintah push. Dan untuk memindahkan data dari tempat tersebut digunakan perintah pop. Sedangkan dalam penyajiannya, stack bisa memakai array atau linked list.

Sebagai contoh, misalkah ada data sebagai berikut : 1 3 5 6, maka data tersebut dapat tersimpan dalam bentuk sebagai berikut :



Dari gambar di atas bisa di lihat bahwa data di dalam stack hanya bisa di masukkan atau di ambil melalui bagian ujung saja/top.

Operasi pada stack melibatkan beberapa fungsi dasar yang dapat dilakukan pada struktur data ini. Berikut adalah beberapa operasi umum pada stack:



a. PUSH

PUSH(S,E) adalah operator yang berfungsi menambahkan elemen E ke Stack. Elemen E akan ditempatkan pada posisi TOP(S). Suatu error akan terjadi jika PUSH dioperasikan pada stack yang sudah mencapai maksimal stack.(Overflow)

b. POP

POP (S) Adalah operator yang berfungsi mengeluarkan atau menghapus elemen TOP(S) dari dalam stack. POP(S) akan mengurangi nilai NOEL(S) dengan 1. Suatu error akan terjadi jika POP(S) dilakukan pada stack yang hampa / kosong. (Underflow)

c. CREATE

CREATE (S) Adalah operator yang menyebabkan Stack S menjadi suatu stack hampa. Jadi NOEL(CREATE(S)) adalah 0 dan TOP(CREATE(S)) adalah tak terdefinisi.

d. ISEMPY

IsEmpty (S) Adalah operator yang berfungsi untuk memeriksa apakah Stack(S) hampa (kosong) atau tidak. Hasil dari operasi ISEMPY adalah Boolean yaitu TRUE jika kosong dan FALSE jika isi.

e. ISFULL

IsFull adalah fungsi yang melakukan pengecekan apakah stack dalam kondisi penuh atau tidak.

f. PEEK

Operasi Peek digunakan untuk mengecek posisi teratas dalam stack.

g. SIZE

Mengembalikan jumlah elemen yang ada dalam tumpukan.

h. CLEAR

Mengosongkan atau menghapus semua elemen dari tumpukan.

i. SEARCH

Mencari keberadaan elemen tertentu dalam tumpukan.

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;
}

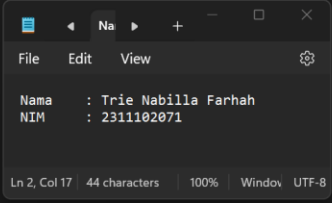
```

Output

```
PS D:\Semester2\Struktur Data\Praktikum6> & 'c:\Users\Asus\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-jihkvuac.bgi' '--stdout=Microsoft-MIEngine-Out-oybspyzb.anx' '--stderr=Microsoft-MIEngine-Error-Sezodv03.xrl' '--pid=Microsoft-MIEngine-Pid-ieh3mdj.wig' '--dbgExe=C:\msys64\ucrt64\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 D:\Semester2\Struktur Data\Praktikum6>
```



Deskripsi Program

Program ini menggunakan array untuk menjalankan stack, yang merupakan struktur data yang menggunakan prinsip Last In, First Out (LIFO). Ini adalah penjelasan tentang cara program berjalan:

- string array buku[5]; Array array buku ukuran 5 untuk menyimpan data stack
- int maksimal = 5, top = 0; kapasitas stack maksimal dan indeks posisi elemen paling atas di stack (jumlah elemen saat ini).
- isFull(): Menentukan apakah stack penuh. Mengembalikan true jika top dan maksimal sama, yang menunjukkan bahwa stack penuh.
- isEmpty(): Menentukan apakah stack kosong. Jika top sama dengan 0, itu berarti stack kosong.
- pushArrayBuku(string data). Menambahkan elemen baru ke stack dengan , Jika stack sudah penuh, ditampilkan pesan bahwa stack sudah penuh. Jika stack tidak penuh, menambahkan elemen baru ke posisi top dan meningkatkan nilai top.
- peekArrayBuku(int posisi): Melihat elemen pada posisi tertentu di atas stack; jika stack kosong, menampilkan pesan bahwa tidak ada data yang bisa dilihat; jika tidak kosong, mengosongkan elemen di posisi top-1 dan mengurangi nilai top.
- popArrayBuku(): Menghapus elemen teratas dari stack. Jika stack kosong, menampilkan pesan bahwa tidak ada data yang bisa dihapus. Jika stack tidak kosong, menghitung indeks yang sesuai dan menampilkan elemen tersebut.
- count Stack: Mengembalikan jumlah elemen dalam stack, yaitu nilai dari top.
- changeArrayBuku(int posisi, string data): Mengubah elemen pada posisi tertentu di stack. Jika posisi lebih besar dari top, akan ditampilkan pesan bahwa posisi melebihi jumlah elemen yang ada. Jika tidak, cari indeks yang sesuai dan ganti elemen dengan data baru.
- destroyArrayBuku(): Menghapus semua elemen di stack, mengosongkan setiap elemen dari atas ke bawah, dan mengatur top menjadi 0.
- cetakArrayBuku(): Mencetak semua elemen di stack dari atas ke bawah. Pesan menunjukkan bahwa data tidak dapat dicetak jika stack kosong.

C. UNGUIDED

Unguided 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.

Source Code

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

using namespace std;

bool kataPalindrome(string kata)
{
    stack<char> tumpukanStack;

    for (char ch : kata)
    {
        tumpukanStack.push(ch);
    }

    for (char ch : kata){
        if (ch != tumpukanStack.top())
        {
            return false;
        }
        tumpukanStack.pop();
    }

    return true;
}

int main()
{
    string masukkan;
    char pilih;
```



```
bool masukkanlanjut = true;

while (masukkanlanjut)
{
    cout << "Masukkan sebuah kata: ";
    cin >> masukkan;

    bool palindrome = kataPalindrome(masukkan);

    if (palindrome)
    {
        cout << "Kata " << masukkan << " Adalah Palindrom."
            << endl;
    }
    else
    {
        cout << "Kata " << masukkan << " Bukan Palindrom."
            << endl;
    }

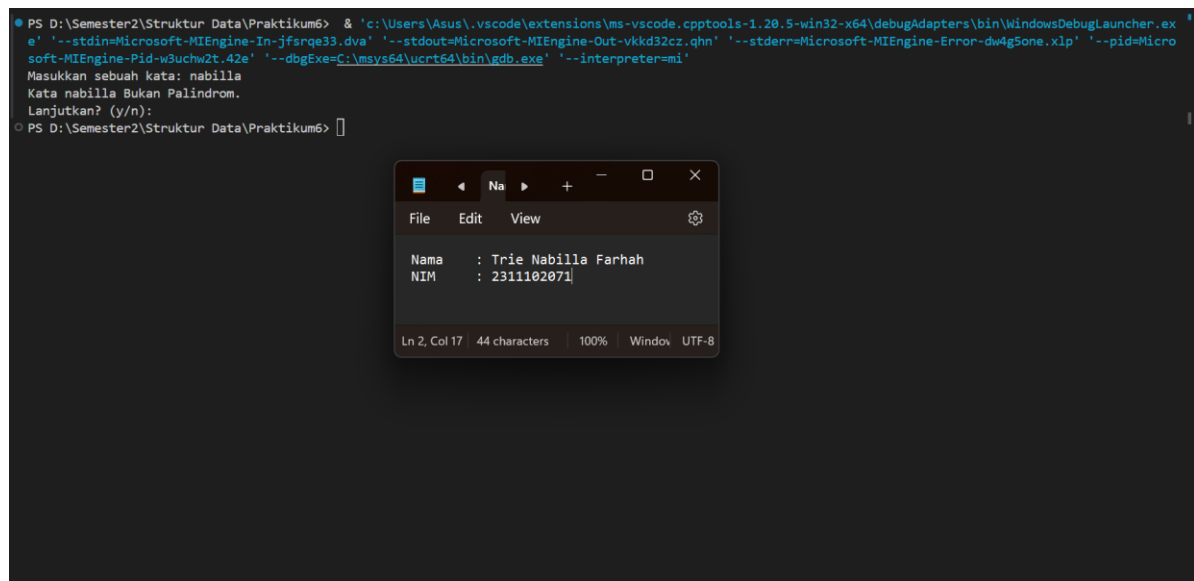
    cout << "Lanjutkan? (y/n): ";
    cin >> pilih;

    if (pilih == 'n' || pilih == 'N')
    {
        masukkanlanjut = false;
    }

    cout << endl;
}

return 0;
}
```

Output



The screenshot shows a terminal window with the following commands and output:

```
PS D:\Semester2\Struktur Data\Praktikum6> & 'c:\Users\Asus\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdout=Microsoft-MIEngine-In-jfsrqe33.dva' '--stdout=Microsoft-MIEngine-Out-vkdd32cz.qhn' '--stderr=Microsoft-MIEngine-Error-dw4g5one.xlp' '--pid=Microsoft-MIEngine-Pid-w3uchw2t.42e' '--dbgexe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Masukkan sebuah kata: nabilla
Kata nabilla Bukan Palindrom.
Lanjutkan? (y/n):
PS D:\Semester2\Struktur Data\Praktikum6>
```

Below the terminal, a small application window titled "Na" is open, displaying the following text:

```
Nama : Trie Nabilla Farhah
NIM  : 2311102071
```

The application window also shows a status bar at the bottom: "Ln 2, Col 17 | 44 characters | 100% | Window | UTF-8".

Deskripsi Program

Program ini menggunakan stack untuk menentukan apakah sebuah kata merupakan palindrom. Palindrom adalah kata yang dapat dibaca dari depan dan belakang.

- Fungsi kata Palindrome menerima string, atau kata, dan mengembalikan nilai boolean, yang menunjukkan apakah kata tersebut merupakan palindrom atau tidak.
- Deklarasi Stack mendeklarasikan sebuah tumpukan bernama tumpukanStack untuk menyimpan karakter kata.
- Mengisi stack Loop untuk (char ch: kata) mengisi stack dengan setiap karakter dalam kata. Karakter pertama dimasukkan ke stack, dan setelah loop selesai, karakter terakhir berada di puncak stack.
- Mengecek palindrom: Loop untuk (char ch: kata) kedua memeriksa setiap karakter dari awal kata dan membandingkannya dengan karakter di puncak stack. Fungsi mengembalikan false jika ada karakter yang tidak cocok; sebaliknya, jika semua karakter cocok, fungsi mengembalikan benar.
- Fungsi main memungkinkan untuk meminta input pengguna dan memeriksa apakah kata-kata tersebut palindrom.
- Variabel deklarasi termasuk masukkan string untuk menyimpan kata yang dimasukkan oleh pengguna, pilih char untuk menyimpan pilihan apakah pengguna ingin melanjutkan atau tidak, dan bool masukkanlanjut = true untuk mengontrol loop.
- loop Input pengguna lingkaran while (lingkaran lanjut) terus berjalan selama lingkaran lanjut bernilai benar.

- Meminta input kata cout << "Masukkan kata: "; meminta pengguna memasukkan kata dan menyimpannya di variabel masukan.
- Memeriksa palindrom berarti menghubungkan kata palindrome dan menyimpan hasilnya di variabel palindrome.
- Menampilkan hasil: Jika palindrome bernilai benar, program menunjukkan bahwa kata itu adalah palindrom, jika bernilai salah, menunjukkan bahwa kata itu bukan palindrom.
- Meminta input untuk melanjutkan, ketikkan cout "Lanjutkan? (y/n): "; meminta pengguna untuk memilih untuk melanjutkan atau tidak. Jika pengguna memasukkan "n" atau "N", lanjutan diatur menjadi tidak benar, sehingga loop berhenti.

Unguided 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?

Source Code

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

string kataterbalik(string input)
{
    stack<char> tumpukankata;
    string kataterbalik = "";

    for (char ch : input)
    {
        tumpukankata.push(ch);
    }

    while (!tumpukankata.empty())
    {
        kataterbalik += tumpukankata.top();
        tumpukankata.pop();
    }

    return kataterbalik;
}

int main()
{
    vector<string> kata;

    string input;
    cout << "-----Ketik 'selesai' jika anda sudah selesai-----"
    << endl;
    cout << "Masukkan kata yang ingin di balik : ";
    getline(cin, input);

    while (input != "selesai")
    {
        kata.push_back(input);

        cout << "Masukkan kata yang ingin di balik : ";
        getline(cin, input);
    }
}
```

```

}

cout << "Kata-kata yang telah diinputkan: " << endl;
for (int i = 0; i < kata.size(); i++)
{
    cout << i + 1 << ". " << kata[i] << endl;
}
cout << endl;

cout << "Kata-kata dalam bentuk terbalik: " << endl;
for (int i = 0; i < kata.size(); i++)
{
    string terbalik = kataterbalik(kata[i]);
    cout << i + 1 << ". " << terbalik << endl;
}
cout << endl;

return 0;
}

```

Output

```

PS D:\Semester2\Struktur Data\Praktikum6> & 'c:\Users\Asus\.vscode\extensions\ms-vscode.cpptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe'
e' '--stdin=Microsoft-MIEngine-In-ylxeuvcq.hlo' '--stdout=Microsoft-MIEngine-Out-cq223rw0.3om' '--stderr=Microsoft-MIEngine-Error-fvgggraf.kio' '--pid=Micro
soft-MIEngine-Pid-egxurydh.egb' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
----Ketik 'selesai' jika anda sudah selesai----
Masukkan kata yang ingin di balik : trie nabilla farhah
Masukkan kata yang ingin di balik : ittelkom purwokerto
Masukkan kata yang ingin di balik : selesai
Kata-kata yang telah diinputkan:
1. trie nabilla farhah
2. ittelkom purwokerto

Kata-kata dalam bentuk terbalik:
1. hahraf alliban eirt
2. otrekowrup mokletti

PS D:\Semester2\Struktur Data\Praktikum6>

```

Na

File Edit View

Nama : Trie Nabilla Farhah
NIM : 2311102071

Ln 2, Col 17 44 characters 100% Window UTF-8

Deskripsi Program

Setelah program menerima beberapa kata dari pengguna, stack digunakan untuk membalikkan setiap kata. Kemudian, program mencetak kata-kata asli dan versi terbalik mereka.

- Fungsi `kataterbalik` menerima string, atau input, dan mengembalikan string yang berisi karakter dari input yang telah dibalik urutannya.

- Deklarasi stack kata sebuah tumpukan kata dideklarasikan untuk menyimpan karakter-karakter dari kata yang akan dibalik.
- Mengisi stack loop untuk (char ch : input) mengisi tumpukan dengan setiap karakter dari input. Karakter pertama dan terakhir akan berada di bagian bawah stack.
- Membalikkan kata: Satu per satu karakter diambil dari puncak stack dan dimasukkan ke string kataterbalik melalui loop while (!tumpukankata.empty()). Dengan cara ini, karakter terakhir dari input diputar ke karakter pertama di kata, dan seterusnya, menghasilkan kata yang dibalik.
- Fungsi main mengumpulkan data pengguna, membalikkan setiap kata, dan menunjukkan hasilnya.
- kumpulan kata "string" untuk menyimpan kata-kata yang dimasukkan oleh pengguna. string input untuk menyimpan input sementara pengguna.
- Petunjuk awal: menunjukkan kepada pengguna untuk mengetik "selesai" jika sudah selesai, memasukkan kata selesai.
- cout "Masukkan kata yang ingin di balik:", pengguna diminta untuk memasukkan sebuah kata.
- getline(cin, input); menerima input pengguna.
- Selama pengguna tidak mengetik "selesai", proses input diulangi while (input!="selesai"); kata yang dimasukkan dimasukkan ke vector kata dengan kata.push_back(input);
- Loop for (int i = 0; i < kata.size(); i++) mencetak setiap kata yang telah diinput oleh pengguna bersama dengan nomornya.

D. KESIMPULAN

Stack atau Tumpukan adalah suatu struktur data yang terbentuk dari barisan hingga yang terurut dari satuan data. Pada Stack, penambahan dan penghapusan elemennya hanya dapat dilakukan pada satu posisi, yaitu posisi akhir stack. Posisi ini disebut Puncak atau Top dari stack.

Stack memiliki beberapa operasi umum, yaitu : Push, Pop, Top, IsEmpty, IsFull, Size, Peek, Clear, Search.

E. REFERENSI

Asisten praktikum, “*Modul 6 Stack*”, learning Management System, 2024

Ranita, S.(2018). ”*Artikel Struktur Data STACK (Tumpukan)*”,jurnal struktur data. 2(1), 1-4.

Suyanto. *Struktur Data* . Diakses pada 18 Mei 2024, dari
[https://sisfo.binadarma.ac.id/upload/materi/8868_SD4-STACK%20\(Tumpukan\).pdf](https://sisfo.binadarma.ac.id/upload/materi/8868_SD4-STACK%20(Tumpukan).pdf)

repository.unikom.*STACK (TUMPUKAN)*. Diakses pada 18 Mei 2024, dari
<https://repository.unikom.ac.id/35662/1/STACK.pdf>