

**LAPORAN PRAKTIKUM STRUKTUR
DATA DAN ALGORITMA**

**MODUL VI
STACK**



Disusun Oleh :

Anisah Syifa Mustika Riyanto
2311102080

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 suatu urutan elemen yang elemennya dapat diambil dan ditambah hanya pada posisi akhir (top) saja. Contoh dalam kehidupan sehari-hari adalah tumpukan piring di sebuah restoran yang tumpukannya dapat ditambah pada bagian paling atas dan jika mengambilnya pun dari bagian paling atas pula.

Ada 2 operasi paling dasar dari stack yang dapat dilakukan, yaitu :

1. Operasi push yaitu operasi menambahkan elemen pada urutan terakhir (paling atas).
2. Operasi pop yaitu operasi mengambil sebuah elemen data pada urutan terakhir dan menghapus elemen tersebut dari stack.

Selain operasi dasar stack (push dan stack), ada lagi operasi lain yang dapat terjadi dalam stack yaitu :

1. Representasi stack dengan array statis
2. Representasi stack dengan linked list

Operasi-operasi stack secara lengkap adalah sebagai berikut:

1. Pendeklarasian stack

Proses pendeklarasian stack adalah proses pembuatan struktur stack dalam memori. Karena stack dapat direpresentasikan dalam 2 cara, maka pendeklarasian stack pun ada 2 yaitu:

- a. Pendeklarasian stack yang menggunakan array statis

```
Const
    Maxstack=.....
Type
    Nama_Stack= array [1..maxstack] of tippedata
Stack : Nama_Stack
Top   : Integer
```

Suatu stack memiliki beberapa bagian yaitu top yang menunjuk posisi data terakhir (top) elemen yang berisi data yang ada dalam stack. Bagian ini lah yang berbentuk array. maks_elemen yaitu variable yang menunjuk maksimal banyaknya elemen dalam stack.

b. Pendeklarasian stack yang menggunakan linked list

Adapun stack yang menggunakan single linked list, hanya memerlukan suatu pointer yang menunjuk ke data terakhir (perhatikan proses di halaman sebelumnya). Setiap elemen linked list mempunyai 2 field yaitu medan data (info) dan pointer sambung (next). Contoh pada single linked list:

```

Type
  Nama_pointer    = ↑simpul
  Simpul          = Record
    <info  : typedata,
      Next : Nama_pointer>
  Endrecord

  Var_pointer : Nama_pointer

```

2. Operasi Push

Operasi push adalah operasi dasar dari stack. Operasi ini berguna untuk menambah suatu elemen data baru pada stack dan disimpan pada posisi top yang akan mengakibatkan posisi top akan berubah. Langkah operasi ini adalah:

a. Operasi push pada stack yang menggunakan array.

Langkah operasi push dalam array adalah dengan :

- Stack dapat ditambah jika stack belum penuh
- Proses push-nya sendiri adalah dengan menambah field top dengan 1, kemudian elemen pada posisi top diisi dengan elemen data baru.

b. Operasi push pada stack yang menggunakan single linked list

Operasi push pada stack yang menggunakan single linked list adalah sama dengan proses tambahawal pada operasi linked list. Langkah-langkahnya adalah: Proses push-nya sendiri adalah dengan cara mengalokasikan suatu elemen linked list (disebut variable baru) dan memeriksa apakah stack kosong / tidak.

3. Operasi Pop

Operasi pop adalah salah satu operasi paling dasar dari stack. Operasi ini berguna untuk mengambil elemen terakhir (top) dan kemudian menghapus elemen tersebut sehingga posisi top akan berpindah. Operasi ini biasanya dibuat dalam bentuk function yang me- return-kan nilai sesuai data yang ada di top.

a. Operasi pop pada stack yang menggunakan array.

Langkah operasi pop pada stack yang menggunakan array adalah

- Stack dapat mengeluarkan elemennya jika stack tidak kosong
- Elemen yang dikeluarkan disimpan pada suatu variable
- Nilai top berkurang 1

b. Operasi pop pada stack yang menggunakan single linked list

Langkah operasi pop pada stack yang menggunakan single linked list adalah sama dengan proses hapusawal pada operasi single linked list. Prosesnya adalah: Periksa apakah stack kosong (isempty), jika kosong maka proses pop tidak bisa dilakukan. Jika stack tidak kosong maka proses pop dijalankan. Proses pop-nya sendiri adalah mengambil elemen yang ditunjuk oleh pointer stack kemudian simpan dalam variable data. Kemudian buat variable pointer bantu yang diisi dengan pointer penunjuk stack yang nantinya akan dihapus dari memori. Kemudian pointer penunjuk stack dipindahkan ke posisi yang ditunjuk oleh field pointer bawah dari variable bantu.

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;
}

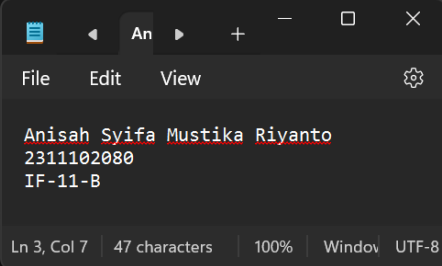
```

Screenshots Output:

```
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> & 'c:\Users\hp151\.vscode\extensions\ms-vscode.cp
ptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-2gwuh5fx.0ch
' '--stdout=Microsoft-MIEngine-Out-aqxn12hd.2tf' '--stderr=Microsoft-MIEngine-Error-nn05q05e.3ia' '--pid=Micros
oft-MIEngine-Pid-otzh0xih.e13' '--dbgExe=C:\Program Files\CodeBlocks\MinGW\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:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> |
```



The screenshot shows a code editor window with a dark theme. The editor displays a list of subjects: "Anisah Syifa", "Mustika Riyanto", "2311102080", and "IF-11-B". The editor has a menu bar with "File", "Edit", and "View". The status bar at the bottom shows "Ln 3, Col 7", "47 characters", "100%", "Window", and "UTF-8".

Deskripsi:

Program mengimplementasikan stack menggunakan array . Stack didefinisikan dengan kapasitas lima elemen (maksimal = 5) yang disimpan dalam array "buku array." Fungsi "isFull()" dan "isEmpty()" menentukan apakah stack penuh atau kosong. Fungsi "pushArrayBuku()" menambahkan elemen baru ke dalam stack jika stack belum penuh, sementara "popArrayBuku()" menghapus elemen teratas dari stack jika stack tidak kosong. Fungsi "peekArrayBuku()" menampilkan elemen di posisi tertentu di atas stack tanpa menghapusnya, dan fungsi-fungsi ini dapat digunakan untuk menambah, menghapus, melihat, mengubah, dan mencetak elemen dalam bagian main.

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.

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

int main()
{
    string a;
    cout << "Masukkan Kalimat: ";
    cin >> a;

    stack<char> stackChar;
    int lengthInput = a.length();

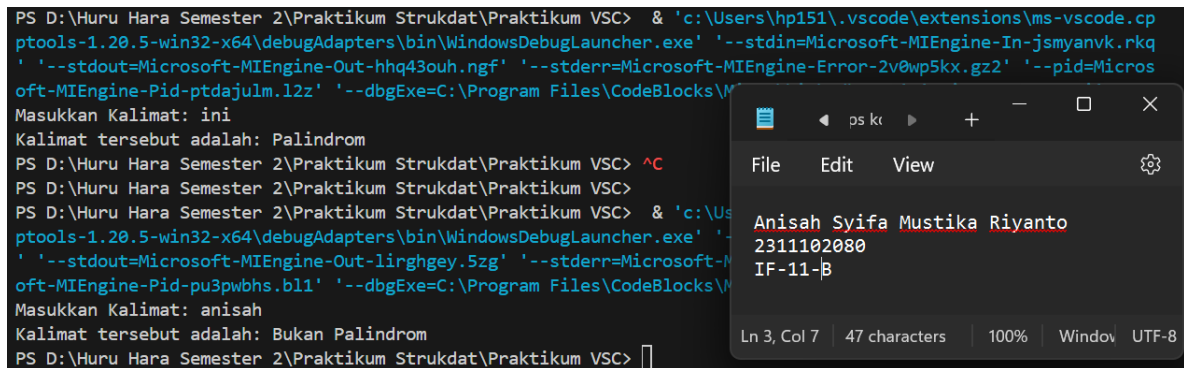
    for (int i = 0; i < lengthInput; ++i)
    {
        stackChar.push(a[i]);
    }

    bool palindrom = true;
    for (int i = 0; i < lengthInput; ++i)
    {
        if (a[i] != stackChar.top())
        {
            palindrom = false;
            break;
        }
        stackChar.pop();
    }

    cout << "Kalimat tersebut adalah: ";
    if (palindrom)
    {
        cout << "Palindrom" << endl;
    }
    else
    {
        cout << "Bukan Palindrom" << endl;
    }

    return 0;
}
```

Screenshots Output:



```
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> & 'c:\Users\hp151\.vscode\extensions\ms-vscode.cp
ptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-jsmyanvk.rkq
' '--stdout=Microsoft-MIEngine-Out-hhq43ouh.ngf' '--stderr=Microsoft-MIEngine-Error-2v0wp5kx.gz2' '--pid=Micros
oft-MIEngine-Pid-ptdajulm.12z' '--dbgExe=C:\Program Files\CodeBlocks\M
Masukkan Kalimat: ini
Kalimat tersebut adalah: Palindrom
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> ^C
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC>
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> & 'c:\Us
ptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '-
' '--stdout=Microsoft-MIEngine-Out-lirhgey.5zg' '--stderr=Microsoft-M
oft-MIEngine-Pid-pu3pwbhs.bl1' '--dbgExe=C:\Program Files\CodeBlocks\M
Masukkan Kalimat: anisah
Kalimat tersebut adalah: Bukan Palindrom
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> []
```

Anisah Syifa Mustika Riyanto
2311102080
IF-11-β

Deskripsi:

Program ini menentukan apakah sebuah kalimat adalah palindrom atau tidak dengan meminta input kalimat dari pengguna, membalikkan kalimat tersebut menggunakan stack, dan kemudian memeriksa apakah hasilnya sama atau tidak. Jika hasilnya sama, maka kalimat tersebut adalah palindrom, jika tidak, maka kalimat tersebut bukan palindrom. Untuk membalikkan kalimat, program ini menggunakan stack sehingga lebih mudah untuk menentukan apakah kalimat tersebut adalah palindrom atau tidak.

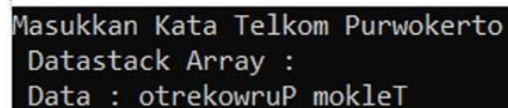
Unguided 2

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

Contoh

Kalimat : Telkom Purwokerto

Hasil : otrekowruP mokleT



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

Source code:

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

int main()
{
    string a;

    // Loop untuk memastikan a terdiri dari minimal 3 kata
    while (true) {
        cout << "Masukkan Kalimat (minimal 3 kata): ";
        getline(cin, a);

        // Hitung jumlah kata
        int wordCount = 0;
```



```

        bool inWord = false;
        for (char c : a) {
            if (isspace(c)) {
                if (inWord) {
                    wordCount++;
                    inWord = false;
                }
            } else {
                inWord = true;
            }
        }
        if (inWord) wordCount++;

        if (wordCount >= 3) {
            break;
        } else {
            cout << "Kalimat harus terdiri dari minimal 3 kata.
Silakan coba lagi." << endl;
        }
    }

    stack<char> stackChar;
    int lengthInput = a.length();

    for (int i = 0; i < lengthInput; ++i) {
        stackChar.push(a[i]);
    }

    cout << "Kalimat terbalik: ";
    for (int i = 0; i < lengthInput; ++i) {
        cout << stackChar.top();
        stackChar.pop();
    }
    cout << endl;

    return 0;
}

```

Screenshots Output:

The screenshot shows a Windows Command Prompt window with the following text:

```

PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC> & 'c:\Us
ptools-1.20.5-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '-
' '--stdout=Microsoft-MIEngine-Out-0e1pew21.jzd' '--stderr=Microsoft-M
oft-MIEngine-Pid-zmapww5v.ma1' '--dbgExe=C:\Program Files\CodeBlocks\N
Masukkan Kalimat (minimal 3 kata): Anisah
Kalimat harus terdiri dari minimal 3 kata. Silakan coba lagi.
Masukkan Kalimat (minimal 3 kata): Anisah Syifa Mustika Riyantoooo
Kalimat terbalik: oooonayir akitsuM afiyS hasinA
PS D:\Huru Hara Semester 2\Praktikum Strukdat\Praktikum VSC>

```

Overlaid on the bottom right is a Notepad++ window with the following text:

```

Anisah Syifa Mustika Riyanto
2311102080
IF-11-B

```

The Notepad++ status bar at the bottom indicates: Ln 3, Col 7 | 47 characters | 100% | Window | UTF-8.

Deskripsi:

Program meminta pengguna untuk memasukkan kata minimal tiga kata. Setelah itu, kalimat tersebut ditampilkan terbalik.

Pertama, program meminta pengguna untuk menulis minimal 3 kata. Jika kurang dari 3 kata program otomatis menampilkan pesan untuk mencoba lagi dan kembali ke menu input awal. Untuk menghitung jumlah kata, program mengiterasi setiap karakter dalam string input dan menggunakan variabel untuk mengetahui apakah karakter tersebut termasuk dalam sebuah kata. Jika jumlah kata mencapai tiga atau lebih, program berhenti meminta input.

Program menyimpan setiap karakter dari kata ke dalam sebuah stack. Kemudian, program mengiterasi stack, mengambil setiap karakter dari atas stack, dan menampilkannya, sehingga menghasilkan kata dalam urutan terbalik. Contohnya, jika pengguna memasukkan kata "Saya makan", hasilnya akan menjadi "nakam ayaS".

Program ini menggunakan loop, pengkondisian, stack untuk memvalidasi input pengguna dan mengubah string sesuai kebutuhan. Hal ini dilakukan dengan menampilkan kalimat dalam urutan terbalik setelah memastikan bahwa kalimat minimal memiliki tiga kata.

D. Kesimpulan

Berikut adalah beberapa poin kesimpulan mengenai stack:

1. Stack adalah struktur data linier yang mengikuti prinsip operasi LIFO (Last In First Out), dimana elemen yang disisipkan terakhir akan menjadi elemen yang diambil pertama kali.
2. Stack memungkinkan manajemen data yang efisien, manajemen fungsi yang efisien.
3. Operasi dasar pada stack adalah operasi push untuk menambahkan elemen pada urutan terakhir dan operasi pop untuk mengambil elemen pada urutan terakhir dan menghapus elemen tersebut dari stack.

E. Referensi

Aliyanto, A. (2011, June). Sistem Pembelajaran Algoritma Stack dan Queue dengan Pendekatan Problem Based Learning untuk Mendukung Pembelajaran Struktur Data. In *Seminar Nasional Aplikasi Teknologi Informasi (SNATI)*.

Dharmayanti, D. (2011). Stack.

Erkamim, E., Abdurrohman, I., Yuliyanti, S., Karim, R., Rahman, A., Admira, T. M. A., & Ridwan, A. (2024). *Buku Ajar Algoritma dan Struktur Data*. PT. Sonpedia Publishing Indonesia.

Mukharil Bachtiar, A. (2012). Stack (pelengkap).

Sumantri, Y. (2001). Pengajaran Berbantuan Komputer Untuk Matakuliah Struktur Data Pada Topik Stack, Queue Dan Linked List.