

# **JOBSHEET**



## ***STACK DAN OPERASI DASAR PADA STACK***

**Oleh:**

**HENDRAWATY, ST., MT.**

**PROGRAM STUDI TEKNIK INFORMATIKA  
JURUSAN TEKNOLOGI INFORMASI DAN KOMPUTER  
POLITEKNIK NEGERI LHOKSEUMAWE  
2020**

## HALAMAN PENGESAHAN INSTITUSI

### STACK DAN OPERASI DASAR PADA STACK

Kegiatan Pengembangan Jobsheet ini Dibiayai dengan Sumber Dana DIPA Politeknik Negeri Lhokseumawe Tahun Anggaran 2020



Mengetahui,  
Ketua Jurusan Teknologi Informasi  
dan Komputer,

Penulis,

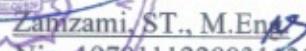
  
Muhammad Arhami, S.Si., M.Kom.  
Nip. 197410292000031001

  
Hendrawaty, ST., MT.  
Nip. 197002261998022001

Mengetahui/Mengesahkan:  
Wakil Direktur Bidang Akademik, Kemahasiswaan, dan  
Alumni

Politeknik Negeri Lhokseumawe,



  
Zamzami, ST., M.Eng.  
Nip. 1979111220031121003

## HALAMAN PENGESAHAN REVIEWER

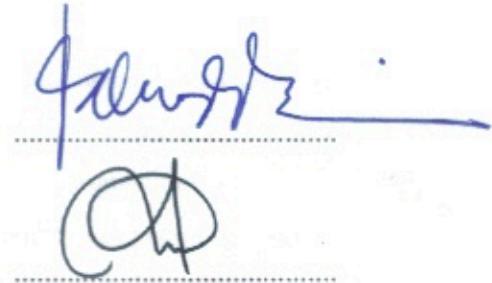
Jobsheet *Stack* dan Operasi Dasar pada *Stack* yang disusun oleh:

Nama : Hendrawaty, ST., MT.  
NIP : 197002261998022001  
Jurusan : Teknologi Informasi dan Komputer

Telah memenuhi syarat-syarat penulisan Jobsheet yang dibiayai dengan sumber dana DIPA Politeknik Negeri Lhokseumawe Tahun Anggaran 2020.

### Reviewer :

1. Salahuddin, ST, M.Cs.  
NIP. 197404242002121001
2. Muhammad Rizka, SST, M. Kom.  
NIP. 198810091015041001



Mengetahui,  
Kepala Pusat Pengembangan  
Pembelajaran dan Penjaminan Mutu



Ir. Heri Mahyar, MT.

NIP. 19621201 198902 1001

Menyetujui,  
Ketua Departemen Pendidikan  
Dan Pengembangan Pembelajaran

Ir. Jufriadi, MT.

NIP. 19641102 199303 1002

**LABORATORIUM: TELEMATIKA**  
**POLITEKNIK NEGERI LHOKSEUMAWE**  
**PENGUJIAN : STACK DAN OPERASI DASAR PADA STACK**

### I. Capaian Praktikum/Kompetensi

- Mahasiswa dapat menjelaskan cara kerja *stack*
- Mahasiswa dapat melakukan operasi *pop* dan *push* pada *stack*.
- Mahasiswa dapat mengimplementasikan *Stack* dalam bentuk program dengan bahasa C.

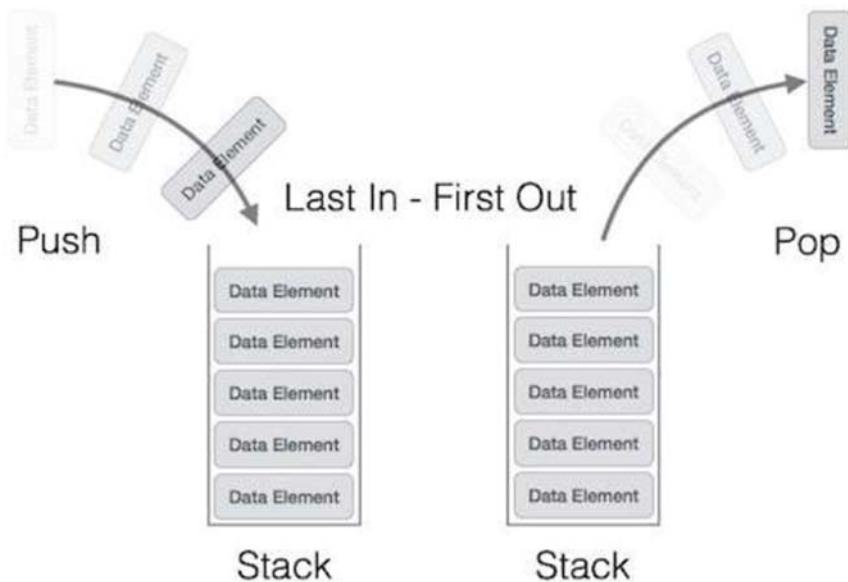
### II. Keselamatan Kerja

- Praktikum hanya dapat dilakukan atas petunjuk dosen pengasuh atau instruktur.
- Pastikan semua peralatan dalam keadaan baik sebelum dan sesudah melakukan praktikum serta menjaga semua peralatan yang digunakan.
- Perlu mematuhi semua peraturan pada laboratorium yang digunakan
- Sebelum melakukan praktikum agar selalu mengawali dengan berdoa

### III. Teori

*Stack* adalah Tipe Data Abstrak (ADT), yang biasa digunakan di sebagian besar bahasa pemrograman. Ini dinamai *stack* karena berperilaku seperti *Stack* (tumpukan) yang sebenarnya di dunia nyata, misalnya - tumpukan kartu atau tumpukan piring, dll. Fitur ini menjadikannya struktur data LIFO. LIFO adalah singkatan dari *Last-in-first-out*. Di sini, elemen yang ditempatkan (dimasukkan atau ditambahkan) terakhir, diakses terlebih dahulu.

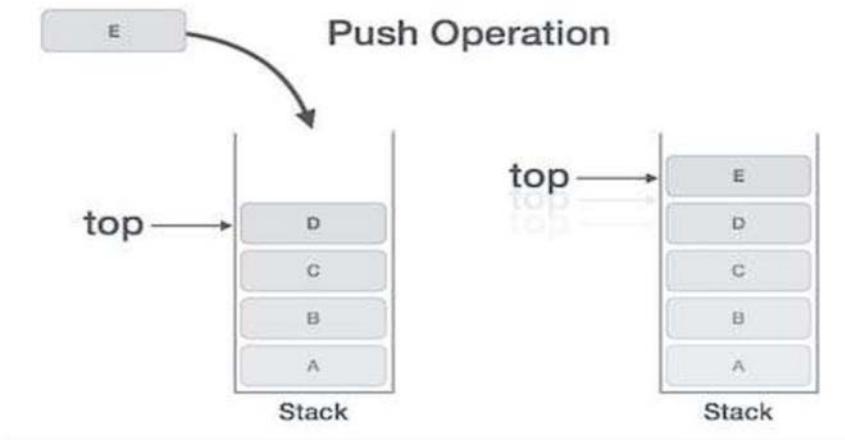
*Stack* dapat diimplementasikan melalui *Array*, *Structure*, *Pointer*, dan *Linked List*. *Stack* dapat berupa ukuran tetap atau mungkin memiliki ukuran yang dinamis. Pada *Job sheet* ini *stack* diimplementasikan menggunakan *array*, yang membuatnya menjadi implementasi *stack* dengan ukuran tetap. Pada Gambar 1. Dapat dilihat ilustrasi dari *Stack*.



Gambar 1. Ilustrasi *stack*

### Operasi dasar pada *stack*

Terdapat dua operasi dasar pada *stack*, yaitu operasi *Push* dan operasi *Pop*. Operasi *Push* adalah proses menempatkan elemen data baru ke *stack*. Gambar 2. Memperlihatkan ilustrasi operasi *push* pada *stack*.



Gambar 2. Ilustrasi operasi *Push* pada *stack*

Operasi *push* melibatkan serangkaian langkah :

- Langkah 1 - Memeriksa apakah *stack* sudah penuh.
- Langkah 2 - Jika *stack* penuh, buat pesan kesalahan (*error*) dan keluar.
- Langkah 3 - Jika *stack* tidak penuh, tambahkan bagian atas (*top*) untuk

menunjuk ruang kosong berikutnya.

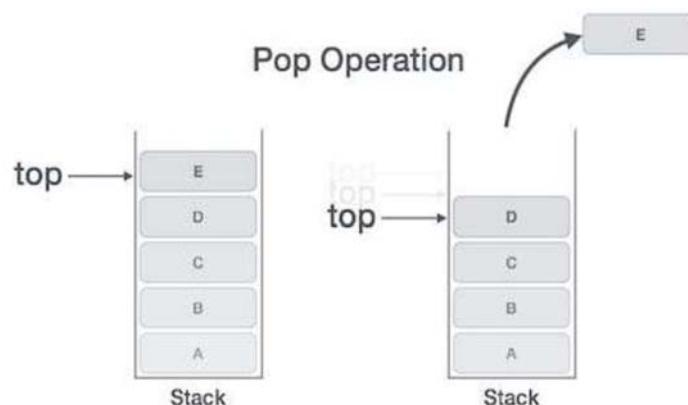
- Langkah 4 - Menambahkan elemen data ke lokasi *stack*, tempat puncak menunjuk.
- Langkah 5 - Mengembalikan kesuksesan (*returns success*).

Algoritma sederhana untuk operasi push dapat dilihat pada Gambar 3.

```
begin procedure push: stack, data
    if stack is full
        return null
    endif
    top ← top + 1
    stack[top] ← data
end procedure
```

Gambar 3. Algoritma operasi *push*

Sedangkan operasi *Pop* (*Pop operation*) adalah proses mengakses konten sambil menghapusnya dari *stack*. Dalam implementasi array operasi *Pop*, elemen data tidak benar-benar dihapus, sebaliknya top dikurangi ke posisi yang lebih rendah di *stack* untuk menunjuk ke nilai berikutnya. Tetapi dalam implementasi *linked-list*, *Pop* sebenarnya menghapus elemen data dan men-deallocate ruang memori. Gambar 4. Memperlihatkan ilustrasi operasi *pop* pada *stack*.



Gambar 4. Ilustrasi operasi *Pop* pada *stack*

Operasi *Pop* melibatkan langkah-langkah berikut :

- Langkah 1 - Memeriksa apakah *stack* kosong.
- Langkah 2 - Jika *stack* kosong, buat kesalahan (pesan *error*) dan keluar (*exit*).
- Langkah 3 - Jika *stack* tidak kosong, akses elemen data di mana puncak (*top*) menunjuk.
- Langkah 4 - Mengurangi nilai *top* dengan 1.
- Langkah 5 - Mengembalikan kesuksesan (*returns success*).

Algoritma sederhana untuk operasi push dapat dilihat pada Gambar 5.

```
begin procedure pop: stack
    if stack is empty
        return null
    endif
    data ← stack[top]
    top ← top - 1
    return data
end procedure
```

Gambar 5. Algoritma operasi *pop*

#### IV. Alat/Bahan

1. Komputer Pribadi (*Personal Computer*)
2. Perangkat Lunak Dev-C++, gcc, vim atau pico
3. Sistem operasi windows atau linux

#### V. Prosedur Praktikum

1. Dengan bahasa C, Buatlah program menggunakan fungsi untuk mengimplementasikan operasi *push* pada *stack* berdasarkan langkah langkah yang telah dijelaskan diatas.
2. Dengan bahasa C, Buatlah program menggunakan fungsi untuk mengimplementasikan operasi *pop* pada *stack* berdasarkan langkah langkah yang telah dijelaskan diatas.

3. Dengan bahasa C, Buatlah program menggunakan fungsi untuk mengimplementasi *stack* berdasarkan langkah-langkah yang telah dijelaskan .

## VI. Data Percobaan

Buat program seperti tugas diatas dan hasil eksekusinya dalam borang berikut ini:

Tabel 1. Program dan Hasil Eksekusi

No Perc.	Program ( <i>Source Code</i> )	Hasil ( <i>Output</i> )
1.		
2.		
3.		
4.		

## VII. Analisa dan Kesimpulan

1. Analisa hasil dari eksekusi program diatas, berikan maksud dari setiap baris program yang diberikan.
2. Dari hasil percobaan yang dilakukan, buat kesimpulan dari masing-masing percobaan diatas.

## VIII. Daftar Pustaka

Jain, H. (2017). Problem Solving in Data Structures & Algorithms Using C. Bhopal, India.

Tutorialspoint. (2016). Data Structures & Algorithms Simply Easy Learning, Tutorial points Ltd, [www.tutorialspoint.com](http://www.tutorialspoint.com).

Das. V. V. (2006) Principles of Data Structures Using C and C++, New Age International Ltd, India.