

# **LAPORAN PRAKTIKUM**

## **MODUL 7**

### **“STACK”**



**Disusun Oleh:**

**MARSELLA DWI JULIANTI (2311104004)**

**SE 07-01**

**Dosen :**

**Yudha Islami Sulistya, S.kom, M.Cs.**

**PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

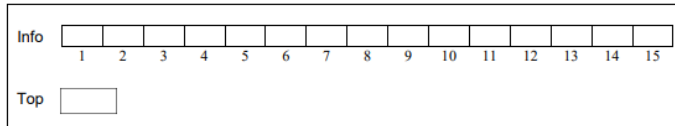
**2024**

## TUGAS PENDAHULUAN MODUL 7 STACK

Anda diminta untuk membuat stack dengan representasi statis, yaitu menggunakan array. Untuk itu buatlah ADT stack pada file "stack.h" dengan struktur berikut:

| Notasi Algoritmik  | Bahasa C++   |
|--|--|
| <pre> type infotype : char type stack : &lt;     info : array[1..15] of infotype     top : integer &gt; </pre> | <pre> typedef int infotype; struct stack {     infotype info[15];     int Top; }; </pre> |

Ilustrasi untuk struktur di atas adalah:



Elemen info[1..15] akan digunakan untuk menampung data char. Sedangkan Top akan digunakan untuk menampung informasi mengenai indeks elemen info yang paling atas. Jadi, Top = 0 artinya stack dalam keadaan kosong. Top = 1, artinya stack berisi satu elemen.

Prosedur dan fungsi yang digunakan adalah

| Notasi Algoritmik   | Bahasa C++                                   |
|---|--|
| <pre> procedure createStack(input/output S : stack) { I.S. sembarang   F.S. terbentuk stack dengan Top = 0 } </pre> | <pre> void createStack(stack &amp;S); </pre> |
| <pre> function isEmpty(S : stack): boolean { Mengembalikan nilai true jika stack kosong } </pre>                    | <pre> bool isEmpty(stack S); </pre>          |
| <pre> function isFull(S : stack): boolean { Mengembalikan nilai true jika stack penuh } </pre>                      | <pre> bool isFull(stack S); </pre>           |

| Notasi Algoritmik   | Bahasa C++   |
|---|--|
| <pre> function isFull(S : stack): boolean { Mengembalikan nilai true jika stack penuh } Kamus: { Tidak ada kamus } Algoritma: if Top(S) = 15 then     → true else     → false { end if } </pre> | <pre> bool isFull(stack S) {     /* Lengkapi kodenya */ } </pre> |

| Notasi Algoritmik   | Bahasa C++  |
|---|---|
| <pre> procedure push(input/output S : stack, input x : infotype) { I.S. mungkin kosong   F.S. menambahkan elemen pada stack dengan     nilai x, Top = Top + 1 } Kamus: { Tidak ada kamus } Algoritma: if isFull(S) = false then     Top(S) ← Top(S) + 1     Info(S)[Top(S)] ← x { end if } </pre> | <pre> void push(stack &amp;S, infotype x) {     /* Lengkapi kodenya */ } </pre> |

| Notasi Algoritmik   | Bahasa C++  |
|---|---|
| <pre> function pop(input/output S : stack): infotype { Mengembalikan nilai infotype yang ada pada indeks   Top, Top = Top - 1 } Kamus: x : infotype Algoritma: x ← Info(S)[Top(S)] Top(S) ← Top(S) - 1 → x </pre> | <pre> int pop(stack &amp;S) {     /* Lengkapi kodenya */ } </pre> |

| Notasi<br>Algoritmik  | Bahasa<br>C++  |
|---|--|
| <pre> <b>procedure</b> printInfo(<b>input</b> S : stack) { I.S. stack mungkin kosong   F.S. Jika stack tidak kosong, maka menampilkan   semua info yang ada pada stack }  Kamus:   i : <b>integer</b> Algoritma:   <b>for</b> i ← Top(S) <b>down to</b> 1 <b>do</b>     <b>output</b>(info(S)[i])   <b>end for</b> </pre> | <pre> <b>void</b> printInfo(stack S) {     /* Lengkapi kodenya */ } </pre> |

Untuk menguji implementasi stack buatlah kode pada "main.cpp" sehingga menghasilkan output berikut ini dengan memanggil function/procedure yang sudah dipakai di atas:

- Digit terakhir NIM MOD 4 sisa 0 :

**Output:**

IFLABJAYA  
JAYA

- Digit terakhir NIM MOD 4 sisa 1 :

**Output:**

HALOBANDUNG  
BANDUNG

- Digit terakhir NIM MOD 4 sisa 2 :

**Output:**

PERCAYADIRI  
DIRI

- Digit terakhir NIM MOD 4 sisa 3 :

**Output:**

STRUKTURDATA  
DATA

Note:

**Output:**

*isi stack awal*

*isi stack sesudah pop*

Semoga Selalu diberi kemudahan^^

Inputan file stack.h :

```

C stack.h  X  G+ stack.cpp  G+ main.cpp
TP > C stack.h > ...
1  #ifndef STACK_H
2  #define STACK_H
3
4  typedef char infotype;
5
6  struct stack {
7      infotype info[15];
8      int Top;
9  };
10
11 void createStack(stack &S);
12 bool isEmpty(stack S);
13 bool isFull(stack S);
14 void push(stack &S, infotype x);
15 infotype pop(stack &S);
16 void printInfo(stack S);
17
18 #endif

```

Inputan file stack.cpp :

```
C stack.h  stack.cpp  main.cpp
TP > stack.cpp > printInfo(stack)
1  #include <iostream>
2  #include "stack.h"
3  using namespace std;
4
5  void createStack(stack &S) {
6      S.Top = 0;
7  }
8
9  bool isEmpty(stack S) {
10     return (S.Top == 0);
11 }
12
13 bool isFull(stack S) {
14     return (S.Top == 15);
15 }
16
17 void push(stack &S, infotype x) {
18     if (!isFull(S)) {
19         S.Top = S.Top + 1;
20         S.info[S.Top] = x;
21     }
22 }
23
24 infotype pop(stack &S) {
25     infotype x = S.info[S.Top];
26     S.Top = S.Top - 1;
27     return x;
28 }
29
30 void printInfo(stack S) {
31     for (int i = S.Top; i >= 1; i--) {
32         cout << S.info[i] << " ";
```

```
33     }
34     cout << endl;
35 }
```

Inputan file main.cpp :

```
C stack.h  stack.cpp  main.cpp
TP > main.cpp > main()
1  #include <iostream>
2  #include "stack.h"
3  using namespace std;
4
5  int main() {
6      stack S;
7      createStack(S);
8
9      // Ganti angka terakhir NIM Anda di sini
10     int nimLastDigit = 0; // Misalnya NIM berakhiran 0
11
12     switch (nimLastDigit % 4) {
13     case 0:
14         // I F L A B J A Y A
15         push(S, 'A');
16         push(S, 'Y');
17         push(S, 'A');
18         push(S, 'J');
19         push(S, 'B');
20         push(S, 'A');
21         push(S, 'L');
22         push(S, 'F');
23         push(S, 'I');
24
25         // Print initial stack
26         printInfo(S);
27
28         // Pop until "JAYA" remains
29         while (S.Top > 4) {
30             pop(S);
31         }
32     }
```

```

33 // Print final stack
34 printInfo(S);
35 break;
36
37 case 1:
38 // H A L O B A N D U N G
39 push(S, 'H');
40 push(S, 'A');
41 push(S, 'L');
42 push(S, 'O');
43 push(S, 'B');
44 push(S, 'A');
45 push(S, 'N');
46 push(S, 'D');
47 push(S, 'U');
48 push(S, 'N');
49 push(S, 'G');
50
51 // Print initial stack
52 printInfo(S);
53
54 // Pop until "BANDUNG" remains
55 while (S.Top > 7) {
56     pop(S);
57 }
58
59 // Print final stack
60 printInfo(S);
61 break;
62

```

Output program:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\07_Stack> ls

Directory: D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\07_Stack

Mode                LastWriteTime         Length Name
----                -
d-----          28/10/2024   13.38             .vscode
d-----          29/10/2024   11.12          GUIDED
d-----          28/10/2024   14.11          output
d-----          30/10/2024   13.52             TP
d-----          29/10/2024   11.39          UNGUIDED

PS D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\07_Stack> cd TP
PS D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\07_Stack\TP> g++ main.cpp stack.cpp -o program
PS D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\07_Stack\TP> ./program
I F L A B J A Y A
J A Y A
PS D:\SEMESTER 3 GANJIL 2024\STRUKTUR DATA\07_Stack\TP>

```

Pada program diatas, terdapat beberapa fungsi utama diantaranya adalah:

- createStack(stack &S), yang menginsialisasi stack dengan mengatur 'Top' menjadi 0, dan mennadakan bahwa stack kosong.
- isEmpty(stack S), dimana ini mengembalikan 'true' jika stack kosong (jika 'Top' sama dengan 0), dan 'false' jika tidak.
- isFull(stack S), mengembalikan 'true' jika stack penuh (jika 'Top' sama dengan 15), dan 'false' jika tidak.
- Push(stack &S, infotype x), yang menambahkan elemen 'x' ke stack jika stack tidak penuh. Jika stack penuh, tidak ada yang dilakukan.
- Pop(stack &S), mengeluarkan elemen teratas dari stack dan mengurangi nilai 'Top'. Jika stack kosong, tidak ada yang dilakukan.
- printInfo(Stack S), yang menampilkan semua elemen dalam stack dari yang teratas hingga yang terbawah.