

Ketentuan Tugas Pendahuluan

1. Tugas Pendahuluan dikerjakan secara **Individu**.
2. TP ini bersifat **WAJIB**, tidak mengerjakan = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
3. Hanya **MENGUMPULKAN** tetapi **TIDAK MENGERJAKAN** = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
4. Deadline pengumpulan TP Modul 7 adalah Senin, 28 Oktober 2024 pukul 06.00 WIB.
5. **TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN**.
6. **DILARANG PLAGIAT (PLAGIAT = E)**.
7. Kerjakan TP dengan jelas agar dapat dimengerti.
8. File diupload di LMS menggunakan format **PDF** dengan ketentuan: **TP\_MOD\_[XX]\_NIM\_NAMA.pdf**
9. **SOAL TEORI WAJIB DIKERJAKAN TULIS TANGAN, TIDAK BOLEH DIKETIK!**

Contoh:

```
int searchNode_130190XXXX (List L, int X);
```

CP (WA):

- Raihan (089638482851)
- Kayyisa (085105303555)
- Abiya (082127180662)
- Rio (081210978384)

**SELAMAT MENGERJAKAN^^**

## 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:

Info

Top

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>

<u>procedure</u> push( <u>input/output</u> S : stack, <u>input</u> x : infotype) { I.S. mungkin kosong F.S. menambahkan elemen pada stack dengan nilai x, Top = Top + 1 }	<b>void</b> push(stack &S, infotype x);
<u>function</u> pop( <u>input/output</u> S : stack): infotype { Mengembalikan nilai pada indeks Top, Top = Top - 1 }	<b>int</b> pop(stack &S);
<u>procedure</u> printInfo( <u>input</u> S : stack) { I.S. stack mungkin kosong F.S. Jika stack tidak kosong, maka menampilkan semua info yang ada pada stack }	<b>void</b> printInfo(stack S);

Gunakan alias sebagai berikut:

Top(S) untuk (S).Top

info(S) untuk (S).info

Buatlah implementasi ADT stack pada file "stack.cpp".

Notasi Algoritmik	Bahasa C++
<u>procedure</u> createStack( <u>input/output</u> S : stack) { I.S. sembarang F.S. terbentuk stack dengan Top = 0 } Kamus: { Tidak ada kamus } Algoritma: Top(S) ← 0	<b>void</b> createStack(stack &S) {  /* Lengkapi kodenya */  }

Notasi Algoritmik	Bahasa C++
<u>function</u> isEmpty(S : stack): <u>boolean</u> { Mengembalikan nilai true jika stackkosong } Kamus: { Tidak ada kamus } Algoritma: if Top(S) = 0 <u>then</u> → <u>true</u> <u>else</u> → <u>false</u> { <u>end if</u> }	<b>bool</b> isEmpty(stack S) {  /* Lengkapi kodenya */  }

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

Notasi Algoritmik	Bahasa C++
<p><u>procedure</u> 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) = <u>false then</u>  Top(S) ← Top(S) + 1  Info(S)[Top(S)] ← x  { end if }</p>	<pre>void push(stack &amp;S, infotype x) {     /* Lengkapi kodenya */ }</pre>

Notasi Algoritmik	Bahasa C++
<p><u>function</u> 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</p>	<pre>int pop(stack &amp;S) {     /* Lengkapi kodenya */ }</pre>

Notasi Algoritmik	Bahasa 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>downto</b> 1 <b>do</b>     <b>output</b>(info(S)[i])   { <b>end for</b> }</pre>	<pre> <b>void</b> printInfo(<b>stack</b> 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*

1. Stackcpp

```
TP > C++ stack.cpp > ...
1  #include "stack.h"
2  #include <iostream>
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++;
20         S.info[S.Top - 1] = x;
21     } else {
22         cout << "Stack penuh!\n";
23     }
24 }
```

```
26  ✓ int pop(stack &S) {  
27  ✓     if (!isEmpty(S)) {  
28         infotype x = S.info[S.Top - 1];  
29         S.Top--;  
30         return x;  
31  ✓     } else {  
32         cout << "Stack kosong!\n";  
33         return -1;  
34     }  
35 }  
36  
37  ✓ void printInfo(stack S) {  
38  ✓     if (isEmpty(S)) {  
39         cout << "Stack kosong\n";  
40  ✓     } else {  
41  ✓         for (int i = S.Top - 1; i >= 0; i--) {  
42             cout << S.info[i] << " ";  
43         }  
44         cout << endl;  
45     }  
46 }
```

## 2. *Stackh*

```
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 int pop(stack &S);
16 void printInfo(stack S);
17
18 #endif
```

### 3. Maincpp

```
TP > C++ main.cpp > main()
1  #include <iostream>
2  #include <cstring> // Untuk fungsi strlen
3  #include "stack.h"
4  #include "stack.cpp"
5  using namespace std;
6
7  void displayStackOutput(stack &S, const char* elements, int popCount) {
8      createStack(S);
9
10     int length = strlen(elements);
11
12     for (int i = length - 1; i >= 0; i--) {
13         push(S, elements[i]);
14     }
15
16     cout << "Isi stack awal:\n";
17     printInfo(S);
18
19     for (int i = 0; i < popCount; i++) {
20         pop(S);
21     }
22
23     cout << "Isi stack setelah beberapa kali pop:\n";
24     printInfo(S);
25 }
```



```
26
27  int main() {
28      stack S;
29      int lastDigit;
30
31      cout << "Masukkan digit terakhir NIM: ";
32      cin >> lastDigit;
33
34      switch (lastDigit % 4) {
35          case 0:
36              displayStackOutput(S, "IFLABJAYA", 5);
37              break;
38          case 1:
39              displayStackOutput(S, "HALOBANDUNG", 4);
40              break;
41          case 2:
42              displayStackOutput(S, "PERCAYADIRI", 7);
43              break;
44          case 3:
45              displayStackOutput(S, "STRUKTURDATA", 8);
46              break;
47          default:
48              cout << "Input tidak valid" << endl;
49      }
50
51      return 0;
52  }
```

```
PS D:\LAPRAK STD\07_stack\TP\output> & .\'main.exe'
Masukkan digit terakhir NIM: 21
Isi stack awal:
H A L O B A N D U N G
Isi stack setelah beberapa kali pop:
B A N D U N G
```

Semoga Selalu diberi kemudahan^^