

### 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 : <u>infotype</u> ) { I.S. mungkin kosong F.S. menambahkan elemen pada stack dengan nilai x, Top = Top + 1 }	<u>void</u> push( <u>stack</u> &S, <u>infotype</u> x);
<u>function</u> pop( <u>input/output</u> S : stack): <u>infotype</u> { Mengembalikan nilai pada indeks Top, Top = Top - 1 }	<u>int</u> pop( <u>stack</u> &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 }	<u>void</u> printInfo( <u>stack</u> 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	<u>void</u> createStack( <u>stack</u> &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> }	<u>bool</u> isEmpty( <u>stack</u> 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(<u>input/output</u> S : stack, <u>input</u> x : <u>infotype</u>)  { 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</u> <u>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(<u>input/output</u> S : stack): <u>infotype</u>  { Mengembalikan nilai infotype yang ada pada indeks Top, Top = Top - 1 }  Kamus:  x : <u>infotype</u>  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> procedure printInfo(input S : stack) { I.S. stack mungkin kosong   F.S. Jika stack tidak kosong, maka menampilkan   semua info yang ada pada stack }  Kamus:   i : integer Algoritma:   for i ← Top(S) downto 1 do     output(info(S)[i])   { end for } </pre>	<pre> void 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*

## KODE PROGRAM

### 1. Stack.h

```
07_Stack > TP > h 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  // Prototipe fungsi
12  void createStack(stack &S);
13  bool isEmpty(stack S);
14  bool isFull(stack S);
15  void push(stack &S, infotype x);
16  int pop(stack &S);
17  void printInfo(stack S);
18
19  #endif
20
```

### 2. Stack.cpp

```
07_Stack > TP > C++ stack.cpp > ...
1  #include "stack.h"
2  #include <iostream>
3  using namespace std;
4
5  // Membuat stack baru dengan Top diinisialisasi 0
6  void createStack(stack &S) {
7      S.Top = 0;
8  }
9
10 // Mengecek apakah stack kosong
11 bool isEmpty(stack S) {
12     return S.Top == 0;
13 }
14
15 // Mengecek apakah stack penuh
16 bool isFull(stack S) {
17     return S.Top == 15;
18 }
19
20 // Menambahkan elemen ke atas stack
21 void push(stack &S, infotype x) {
22     if (!isFull(S)) {
23         S.Top++;
24         S.info[S.Top - 1] = x;
25     } else {
26         cout << "Stack penuh!\n";
27     }
28 }
```

```
29
30 // Menghapus elemen dari atas stack
31 int pop(stack &S) {
32     if (!isEmpty(S)) {
33         infotype x = S.info[S.Top - 1];
34         S.Top--;
35         return x;
36     } else {
37         cout << "Stack kosong!\n";
38         return -1;
39     }
40 }
41
42 // Menampilkan semua elemen dalam stack dari atas
43 void printInfo(stack S) {
44     if (isEmpty(S)) {
45         cout << "Stack kosong\n";
46     } else {
47         for (int i = S.Top - 1; i >= 0; i--) {
48             cout << S.info[i] << " ";
49         }
50         cout << endl;
51     }
52 }
```

### 3. Main.cpp

```
07_Stack > TP > C++ main.cpp > main()
1  #include "stack.h"
2  #include <iostream>
3  using namespace std;
4
5  void testStack(stack &S, const string &data) {
6      // Memasukkan data ke stack
7      for (char c : data) {
8          push(S, c);
9      }
10
11     // Menampilkan isi stack awal
12     cout << "Isi stack awal: ";
13     printInfo(S);
14
15     // Mengeluarkan beberapa elemen dari stack
16     for (size_t i = 0; i < data.size() / 2; i++) {
17         pop(S);
18     }
19
20     // Menampilkan isi stack setelah beberapa elemen di-pop
21     cout << "Isi stack setelah pop: ";
22     printInfo(S);
23 }
24
```

```
int main() {
    stack S;
    createStack(S);

    int nim_last_digit;
    cout << "Masukkan digit terakhir NIM: ";
    cin >> nim_last_digit;

    switch (nim_last_digit % 4) {
        case 0:
            cout << "Output:\nI F L A B J A Y A J A Y A\n";
            testStack(S, "IFLABJAYA");
            break;
        case 1:
            cout << "Output:\nH A L O B A N D U N G B A N D U N G\n";
            testStack(S, "HALOBANDUNG");
            break;
        case 2:
            cout << "Output:\nP E R C A Y A D I R I D I R I\n";
            testStack(S, "PERCAYADIRI");
            break;
        case 3:
            cout << "Output:\nS T R U K T U R D A T A D A T A\n";
            testStack(S, "STRUKTURDATA");
            break;
        default:
            cout << "Digit terakhir NIM tidak valid.\n";
            break;
    }

    return 0;
}
```

### OUTPUT PROGRAM

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS COMMENTS
g++ -o stack_program main.cpp stack.cpp
>
pwsh TP main ?4 ~1 -2 ls 145ms
./stack_program
Masukkan digit terakhir NIM: 26
Output:
P E R C A Y A D I R I D I R I
Isi stack awal: I R I D A Y A C R E P
Isi stack setelah pop: Y A C R E P
Output:
P E R C A Y A D I R I D I R I
Isi stack awal: I R I D A Y A C R E P
Isi stack setelah pop: Y A C R E P
Isi stack awal: I R I D A Y A C R E P
Isi stack setelah pop: Y A C R E P
pwsh TP main ?4 ~1 -2 2s 685ms
```

Semoga Selalu diberi kemudahan^^