

# **LAPORAN PRAKTIKUM STRUKTUR DATA**

## **MODUL I PENGENALAN CODE BLOCKS**



**Disusun Oleh :**

NAMA : Jauza Rasyiq Hernanta

NIM : 103112430033

**Dosen**

WAHYU ANDI SAPUTRA

**PROGRAM STUDI STRUKTUR DATA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

### **A. Dasar Teori**

Stack merupakan salah satu bentuk struktur data dimana prinsip operasi yang digunakan seperti tumpukan. Seperti halnya tumpukan, elemen yang bisa diambil terlebih dahulu adalah elemen yang paling atas, atau elemen yang pertama kali masuk, prinsip ini biasa disebut LIFO (Last In First Out).

### **B. Guided (berisi screenshot source code & output program disertai penjelasannya)**

Guided 1

Main.cpp

```
Modul 7 > main.cpp > main()
1  #include <iostream>
2  #include "stack.h"
3  #include "stack.cpp"
4  using namespace std;
5
6  int main() {
7      cout << "Hello world!" << endl;
8
9      Stack S;
10     CreateStack(S);
11
12     push(S, 3);
13     push(S, 4);
14     push(S, 8);
15     pop(S);
16     push(S, 2);
17     push(S, 3);
18     pop(S);
19     push(S, 9);
20
21     printInfo(S);
22
23     cout << "balik stack" << endl;
24     balikStack(S);
25     printInfo(S);
26
27     return 0;
28 }
29
```

Queue.cpp

Modul 7 > stack.cpp > balikStack(Stack &s)

```
1  #include <iostream>
2  #include "stack.h"
3  using namespace std;
4
5  void CreateStack(Stack &s) {
6      s.top = -1;
7  }
8
9  void push(Stack &s, infotype x) {
10     if (s.top < MAX - 1) {
11         s.top++;
12         s.info[s.top] = x;
13     }
14 }
15
16 infotype pop(Stack &s) {
17     infotype x = -1;
18     if (s.top != -1) {
19         x = s.info[s.top];
20         s.top--;
21     }
22     return x;
23 }
24
25 void printInfo(Stack s) {
26     cout << "[TOP] ";
27     for (int i = s.top; i >= 0; i--) {
28         cout << s.info[i] << " ";
29     }
30     cout << endl;
31 }
32
33 void balikStack(Stack &s) {
34     int i = 0;
35     int j = s.top;
36     while (i < j) {
37         infotype temp = s.info[i];
38         s.info[i] = s.info[j];
39         s.info[j] = temp;
40         i++;
41         j--;
42     }
43 }
```

## Queue.h

```
Modul 7 > C stack.h > ...
1  #ifndef STACK_H
2  #define STACK_H
3
4  #define MAX 20
5
6  typedef int infotype;
7
8  struct Stack {
9      infotype info[MAX];
10     int top;
11 };
12
13 // PRIMITIVE
14 void CreateStack(Stack &s);
15 void push(Stack &s, infotype x);
16 infotype pop(Stack &s);
17 void printInfo(Stack s);
18 void balikStack(Stack &s);
19
20 #endif
21
```

## Screenshots Output

```
PS D:\Praktikum Struktur Data\Laporan Praktikum\Modul 7> cd
Hello world!
[TOP] 9 2 4 3
balik stack
[TOP] 3 4 2 9
PS D:\Praktikum Struktur Data\Laporan Praktikum\Modul 7>
```

## Guided 2

Main.cpp

```

Modul 7 > main.cpp > ...
1  #include <iostream>
2  #include "stack.h"
3  using namespace std;
4
5  int main() {
6      cout << "Hello world!" << endl;
7
8      Stack S;
9      CreateStack(S);
10
11      pushAscending(S, 3);
12      pushAscending(S, 4);
13      pushAscending(S, 8);
14      pushAscending(S, 2);
15      pushAscending(S, 3);
16      pushAscending(S, 9);
17
18      printInfo(S);
19
20      cout << "balik stack" << endl;
21      balikStack(S);
22      printInfo(S);
23
24      return 0;
25  }
26

```

Stack.h

```

Modul 7 > C stackh > ...
1  #ifndef STACK_H
2
3
4  #define MAX 20
5  typedef int infotype;
6
7  struct Stack {
8      infotype info[MAX];
9      int top;
10 };
11
12 void CreateStack(Stack &s);
13 void push(Stack &s, infotype x);
14 infotype pop(Stack &s);
15 void printInfo(Stack s);
16 void balikStack(Stack &s);
17
18 /* TAMBAHAN */
19 void pushAscending(Stack &s, infotype x);
20
21 #endif
22

```

Stack.cpp

```

Modul 7 > stack.cpp > printInfo(Stack)
1  #include <iostream>
2  #include "stack.h"
3  using namespace std;
4
5  void CreateStack(Stack &s) {
6      s.top = -1;
7  }
8
9  void push(Stack &s, infotype x) {
10     if (s.top < MAX - 1) {
11         s.top++;
12         s.info[s.top] = x;
13     }
14 }
15
16 infotype pop(Stack &s) {
17     infotype x = -1;
18     if (s.top != -1) {
19         x = s.info[s.top];
20         s.top--;
21     }
22     return x;
23 }
24
25 void printInfo(Stack s) {
26     cout << "[TOP] ";
27     for (int i = s.top; i >= 0; i--) {
28         cout << s.info[i] << " ";
29     }
30     cout << endl;
31 }
32
33 void balikStack(Stack &s) {
34     int i = 0, j = s.top;
35     while (i < j) {
36         infotype temp = s.info[i];
37         s.info[i] = s.info[j];
38         s.info[j] = temp;
39         i++; j--;
40     }
41 }
42
43
44 void pushAscending(Stack &s, infotype x) {
45     if (s.top == -1) {
46         push(s, x);
47         return;
48     }
49
50     Stack temp;
51     CreateStack(temp);
52
53     while (s.top != -1 && s.info[s.top] > x) {
54         push(temp, pop(s));
55     }
56
57     push(s, x);
58
59     while (temp.top != -1) {
60         push(s, pop(temp));
61     }
62 }
63

```

Output

```
; if ($?) { .\main }  
Hello world!  
[TOP] 9 8 4 3 3 2  
balik stack  
[TOP] 2 3 3 4 8 9  
PS D:\Praktikum Struktukr Data\Laporan Praktikum\Modul 7> |
```

Deskripsi : Kondisi ini setelah menambahkan prosedur pushAscending yang berfungsi memasukkan elemen ke dalam stack dengan menjaga urutan data ascending dari bawah ke atas. Saat data ditampilkan dari TOP, elemen terlihat dalam urutan menurun, dan setelah dilakukan balikStack, urutan menjadi ascending.