

# **LAPORAN PRAKTIKUM**

## **Modul 7**

### **Stack**



**Disusun Oleh:**

**Esuardo bagus prima Julian 2311104025**

**S1SE-07-01**

**Dosen :**

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

**PROGRAM STUDI S1 SOFTWARE ENGINEERING**

**FAKULTAS INFORMATIKA**

**TELKOM UNIVERSITY PURWOKERTO**

**2024**

## I. TP

### 1. stack.h

pada stack.h kita mendefenisikan stuktur data stack dengan tipe elem karakter dan menyediakan fungsi untuk menginisialisasi, memeriksa status kosong atau penuh, push, pop, dan menampilkan elemen yang ada dalam stack.

```
07_Stack > TP > C stack.h > stack > Top
1  ✓ #ifndef STACK_H_INCLUDED
2  ✓ #define STACK_H_INCLUDED
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
19
20
```

### stack.cpp

Mengimplementasikan fungsi-fungsi operasi dasar pada stuktur data stack yang didefinisikan dalam header stack.h, di mana createStack digunakan untuk menginisialisasi stack, isFull dan IsEmpty untuk mengecek stack, push menambahkan elemen ke dalam stack, pop untuk menghapus elemen dari stack dan mengembalikan nilai teratasnya, dan printInfo untuk menampilkan elemen stack.

```
07_Stack > TP > C stack.cpp > isFull(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++;
20         S.info[S.Top] = x;
21     }
22 }
23
24 infotype pop(stack &S) {
25     if (!isEmpty(S)) {
26         infotype x = S.info[S.Top - 1];
27         S.Top--;
28         return x;
29     }
30     return -1;
31 }
32
33 void printInfo(stack S) {
34     for (int i = S.Top - 1; i >= 0; i--) {
35         cout << S.info[i] << " ";
36     }
37     cout << endl;
38 }
39
```

## main.cpp

Program utama yang menggunakan stack untuk menyimpan karakter berdasarkan hasil operasi modulo digit terakhir nim, di mana program menginisialisasi stack sesuai dengan nilai mod, menampilkan isi awal stack dan setelah pop.

```
C_stackh U  C_stack.cpp U  C_main.cpp ...TP U X  C_main.cpp
07_Stack > TP > C_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      int lastDigitNIM = 0;
9      int mod = lastDigitNIM % 4;
10
11      if (mod == 0) {
12          push(S, 'A');
13          push(S, 'Y');
14          push(S, 'A');
15          push(S, 'J');
16          push(S, 'B');
17          push(S, 'A');
18          push(S, 'L');
19          push(S, 'F');
20          push(S, 'I');
21      } else if (mod == 1) {
22          push(S, 'G');
23          push(S, 'N');
24          push(S, 'U');
25          push(S, 'D');
26          push(S, 'N');
27          push(S, 'A');
28          push(S, 'B');
29          push(S, 'O');
30          push(S, 'L');
31          push(S, 'A');
32          push(S, 'H');
33      } else if (mod == 2) {
34          push(S, 'I');
35          push(S, 'R');
36          push(S, 'I');
37          push(S, 'D');
38          push(S, 'A');
39          push(S, 'Y');
40          push(S, 'A');
41          push(S, 'C');
42          push(S, 'R');
43          push(S, 'E');
44          push(S, 'P');
45      } else if (mod == 3) {
46          push(S, 'A');
47          push(S, 'T');
48          push(S, 'A');
49          push(S, 'D');
50          push(S, 'R');
51          push(S, 'U');
52          push(S, 'T');
53          push(S, 'K');
54          push(S, 'U');
55          push(S, 'R');
56          push(S, 'T');
57          push(S, 'S');
58      }
59
60      cout << "Isi awal Stack:" << endl;
61      printInfo(S);
62
63      cout << "Isi setelah pop:" << endl;
64      for (int i = 0; i < 8; i++) {
65          pop(S);
66      }
67      printInfo(S);
68
69      return 0;
70  }
```

## Output:

```
• Isi awal Stack:
S T R U K T U R D A T A
Isi setelah pop:
D A T A
```

## II. UNGUIDED

1. Program di bawah digunakan untuk mengecek kata yang dimasukan termasuk palindrom atau bukan (kata yang sama pada huruf depan dan belakangnya). Kata tersebut disimpan pada stack untuk membandingkan karakter satu per satu. Jika semua karakter dari awal hingga akhir sesuai yang ada dalam stack, maka fungsi akan mengembalikan true di mana ini menunjukkan bahwa kata tersebut adalah palindrom.

```
07_Stack > UNGUIDED > C SOAL_01.cpp > isPalindrome(const string &)\n1  #include <iostream>\n2  #include <stack>\n3  using namespace std;\n4\n5  bool isPalindrome(const string& kalimat) {\n6      stack<char> stackKarakter;\n7      string kalimatString = \"\";\n8\n9      for (char karakter : kalimat) {\n10         if ((karakter >= 'A' && karakter <= 'Z') || (karakter >= 'a' && karakter <= 'z')) {\n11             if (karakter >= 'A' && karakter <= 'Z') {\n12                 karakter += 32;\n13             }\n14             kalimatString += karakter;\n15         }\n16     }\n17\n18     for (char karakter : kalimatString) {\n19         stackKarakter.push(karakter);\n20     }\n21\n22     for (char karakter : kalimatString) {\n23         if (karakter != stackKarakter.top()) {\n24             return false;\n25         }\n26         stackKarakter.pop();\n27     }\n28\n29     return true;\n30 }\n31\n32 int main() {\n33     string kalimatInput;\n34     cout << \"Masukkan kalimat: \";\n35     getline(cin, kalimatInput);\n36\n37     if (isPalindrome(kalimatInput)) {\n38         cout << \"Kalimat tersebut adalah palindrom.\" << endl;\n39     } else {\n40         cout << \"Kalimat tersebut bukan palindrom.\" << endl;\n41     }\n42\n43     return 0;\n44 }\n45
```

Output:

```
PS D:\\KULIAH\\Semester 3\\Struktur Data\\Praktikum - Repo\\07_Stack\\UNGUIDED\\output> cd 'd:\\KULIAH\\Semester 3\\Struktur Data\\Praktikum - Repo\\07_Stack\\UNGUIDED\\output' & .\\SOAL_01.exe\nMasukkan kalimat: Ini\nKalimat tersebut adalah palindrom.\n\nPS D:\\KULIAH\\Semester 3\\Struktur Data\\Praktikum - Repo\\07_Stack\\UNGUIDED\\output> .\\SOAL_01.exe\nMasukkan kalimat: Telkom\nKalimat tersebut bukan palindrom.\n\nPS D:\\KULIAH\\Semester 3\\Struktur Data\\Praktikum - Repo\\07_Stack\\UNGUIDED\\output>
```

2. Program ini membalikan setiap kata yang kita masukan. Fungsi reverseWord akan menerima sebuah kata dan membalikan urutannya menggunakan stack. Setiap karakter yang ada dalam stack tersebut akan diambil satu per satu untuk membentuk kata yang sudah dibalik. Fungsi reverseSentence memisahkan kalimat menjadi kata-kata menggunakan perulangan untuk memeriksa setiap karakter dan menambahkan karakter ke dalam stack jika ditemukan spasi. Setelah semua kata terbalik, kata tersebut dikeluarkan dari stack dan dicetak dengan urutan yang terbalik.

```

07_Stack > UNGUIDED > SOAL_02.cpp > reverseSentence(const string &)
1  #include <iostream>
2  #include <stack>
3  using namespace std;
4
5  string reverseWord(const string& word) {
6      stack<char> stackChar;
7      string reversedWord;
8
9      for (char ch : word) {
10         stackChar.push(ch);
11     }
12
13     while (!stackChar.empty()) {
14         reversedWord += stackChar.top();
15         stackChar.pop();
16     }
17
18     return reversedWord;
19 }
20
21 void reverseSentence(const string& sentence) {
22     stack<string> stackString;
23     string word;
24
25     for (int i = 0; i <= sentence.length(); ++i) {
26         if (i == sentence.length() || sentence[i] == ' ') {
27             if (!word.empty()) {
28                 stackString.push(reverseWord(word));
29                 word.clear();
30             }
31             } else {
32                 word += sentence[i];
33             }
34         }
35
36         while (!stackString.empty()) {
37             cout << stackString.top() << " ";
38             stackString.pop();
39         }
40         cout << endl;
41     }
42
43     int main() {
44         string kalimat;
45         cout << "Masukan kalimat: ";
46         getline(cin, kalimat);
47
48         cout << "Kalimat setelah dibalik: ";
49         reverseSentence(kalimat);
50
51         return 0;
52     }
53

```

Output:

```

Masukan kalimat: Telkom keren
Kalimat setelah dibalik: nerek mokleT

```