

LAPORAN PRAKTIKUM

PERTEMUAN 7

Stack



Nama :

Andika Rifki Pratama (2311104011)

Dosen :

Yudha Islami Sulistya,
S.Kom.,M.Kom.

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

B. Soal Tugas Pendahuluan

1. Code:

Stack.h

```
TP > C stack.h > stack
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 void createStack_2311104011(stack &S);
11 void createStack_2311104011(stack &S);
12 bool isEmpty_2311104011(stack S);
13 bool isFull_2311104011(stack S);
14 void push_2311104011(stack &S, infotype x);
15 infotype pop(stack &S);
16 void printStack_2311104011(stack S);
17
18 #endif
```

Pada stack.h terdapat beberapa deklarasi dari subclass yang digunakan, mulai dari createstack sampai dengan printstack. Selain itu juga terdapat struct untuk membuat stack baru.

Stack.cpp

```
TP > C stack.cpp > pop(stack &S)
1  #include <iostream>
2  #include "stack.h"
3
4  void createStack_2311104011(stack &S) {
5      S.Top = 0;
6  }
7
8  bool isEmpty_2311104011(stack S) {
9      return S.Top == 0;
10 }
11
12 bool isFull_2311104011(stack S) {
13     return S.Top == 15;
14 }
15
16 void push_2311104011(stack &S, infotype x) {
17     if (!isFull_2311104011(S)) {
18         S.info[S.Top - 1] = x;
19         S.Top++;
20     }
21 }
22
23 infotype pop(stack &S) {
24     if (!isEmpty_2311104011(S)) {
25         infotype x = S.info[S.Top - 1];
26         S.Top--;
27         return S.info[S.Top];
28     }
29     return '\0';
30 }
31
32 void printStack_2311104011(stack S) {
33     for (int i = S.Top - 1; i >= 0; i--) {
34         std::cout << S.info[i] << " ";
35     }
36     std::cout << std::endl;
37 }
```

Pada file ini terdapat pendefinisian fungsi operasi dari tiap method yang sudah di deklarasikan pada stack.h. pada file ini createStack digunakan untuk membuat stack

baru, isEmpty dan isFull untuk memeriksa apakah stack kosong, push dan pop digunakan untuk menambahkan dan menghapus variable dalam stack, dan terakhir terdapat fungsi printStack yang mana digunakan untuk memberikan output stack.

Main.cpp

```
TP > main.cpp > main()
1  #include <iostream>
2  #include "stack.h"
3
4  int main() {
5      stack S;
6      createStack_2311104011(S);
7      push_2311104011(S, '!');
8      push_2311104011(S, 'G');
9      push_2311104011(S, 'N');
10     push_2311104011(S, 'U');
11     push_2311104011(S, 'D');
12     push_2311104011(S, 'N');
13     push_2311104011(S, 'A');
14     push_2311104011(S, 'B');
15     push_2311104011(S, 'O');
16     push_2311104011(S, 'L');
17     push_2311104011(S, 'A');
18     push_2311104011(S, 'H');
19
20     std::cout << "Isi awal Stack:" << std::endl;
21     printStack_2311104011(S);
22
23     std::cout << "Isi setelah pop:" << std::endl;
24     for (int i = 0; i < 5; i++) {
25         pop(S);
26     }
27     printStack_2311104011(S);
28
29     return 0;
30 }
```

Pada file ini terdapat pemanggilan fungsi push untuk memasukan variable yang dibutuhkan berupa kata HALO BANDUNG, setelahnya terdapat fungsi untuk emmberikan output kata yang ada di stack, setelahnya terdapat fungsi untuk menghaspus beberapa huruf yang sesuai dengan kebutuhan dengan fungsi pop. Setelah di pop maka program akan memberikan outputan berupa kalimat setelah beberapa huruf dihapus.

Output:

```
PS D:\Campuss\SMT 3\Praktikum Struktur Data\07_Stack\TP> g++ main.cpp stack.cpp -o tepe -mconsole
>>
PS D:\Campuss\SMT 3\Praktikum Struktur Data\07_Stack\TP> ./tepe
Isi awal Stack:
H A L O B A N D U N G
Isi setelah pop:
B A N D U N G
PS D:\Campuss\SMT 3\Praktikum Struktur Data\07_Stack\TP> 
```

Unguided :

1. Buatlah program untuk menentukan apakah kalimat tersebut yang diinputkan dalam program stack adalah palindrom/tidak. Palindrom kalimat yang dibaca dari depan dan belakang sama. Jelaskan bagaimana cara kerja programnya.

```

Unguided_True > G no1.cpp > ...
1  #include <iostream>
2  #include <stack>
3  #include <string>
4
5  using namespace std;
6
7  bool isPalindrome_2311104011(const string& sentence) {
8      stack<char> s;
9      string cleanedSentence;
10
11     for (char ch : sentence) {
12         if (isalpha(ch)) {
13             ch = tolower(ch);
14             cleanedSentence += ch;
15             s.push(ch);
16         }
17     }
18
19     string reversedSentence;
20     while (!s.empty()) {
21         reversedSentence += s.top();
22         s.pop();
23     }
24
25     return cleanedSentence == reversedSentence;
26 }
27
28 int main() {
29     string sentence;
30
31     cout << "Masukan Kalimat: ";
32     getline(cin, sentence);
33
34     if (isPalindrome_2311104011(sentence)) {
35         cout << "Kalimat tersebut adalah: Palindrom" << endl;
36     } else {
37         cout << "Kalimat tersebut adalah: Bukan Palindrom" << endl;
38     }
39
40     return 0;

```

Pada program ini terdapat Boolean untuk memeriksa apakah kata yang dimasukan merupakan kalimat Palindrome atau tidak caranya adalah dengan cara memeriksa apakah kalimat yang dimaksudkan merupakan alphabet atau tidak, apabila iya, kemudian kode akan mengubah kata menjadi huruf kecil, dan kemudian memasukan kalimat kepada stack yang sudah dibuat di awal. Kemudian terdapat main, disini digunakan untuk memberikan output berupa apakah kalimat tersebut adalah palindrom atau bukan.

Output:

```

PS D:\Campuss\SMT 3\Praktikum Struktur Data\07_Stack\Unguided_True> g++ no1.cpp
PS D:\Campuss\SMT 3\Praktikum Struktur Data\07_Stack\Unguided_True> ./a
Masukan Kalimat: kasur rusak
Kalimat tersebut adalah: Palindrom
PS D:\Campuss\SMT 3\Praktikum Struktur Data\07_Stack\Unguided_True>

```

2. Code:

```

Unguided_True> g++ no2.cpp & main()
1  #include <iostream>
2  #include <stack>
3  #include <string>
4  #include <sstream>
5
6  using namespace std;
7
8  string reverseWords_2311104011(const string& sentence) {
9      stringstream ss(sentence);
10     string word, result;
11
12     while (ss >> word) {
13         stack<char> s;
14
15         for (char ch : word) {
16             s.push(ch);
17         }
18
19         string reversedWord;
20         while (!s.empty()) {
21             reversedWord += s.top();
22             s.pop();
23         }
24         result += reversedWord + " ";
25     }
26     return result;
27 }
28
29 int main() {
30     string sentence;
31
32     cout << "Masukkan Kalimat: ";
33     getline(cin, sentence);
34
35     string reversedSentence = reverseWords_2311104011(sentence);
36     cout << "Data stack Array: " << endl;
37     cout << "Data : " << reversedSentence << endl;
38
39     return 0;
40 }
41
42

```

Pada program ini terdapat method reverseword. Pada awal code terdapat deklarasi word, result, dan sentence. Setelahnya ada perulangan untuk ss lebih besar dari kata. Maka akan membuat stack baru, kemudian terdapat fungsi untuk meletakkan kata paling akhir di tempat sementara, kemudian akan dimasukan Kembali supaya terbentuk kata yang ditulis secara terbalik.

Setelahnya ada print.

Output :

```

PS D:\Campuss\SMT 3\Praktikum Struktur Data\07_Stack\Unguided_True> g++ no2.cpp -o reverse -mconsole
>>
PS D:\Campuss\SMT 3\Praktikum Struktur Data\07_Stack\Unguided_True> ./reverse
Masukkan Kalimat: Burung Terbang
Data stack Array:
Data : gnuruB gnabreT
PS D:\Campuss\SMT 3\Praktikum Struktur Data\07_Stack\Unguided_True>

```