

LAPORAN PRAKTIKUM
PERTEMUAN VII
STACK



Nama :

Mei sari mantiantini (2311104012)

Dosen :

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

PROGRAM STUDI S1 REKAYASA PERANGKAT LUNAK
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2024

Stack.h

```
TP > C stack.h
1  #include <iostream>
2  using namespace std;
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);
```

Stack.cpp

```
TP > G stack.cpp > ...
1  #include "stack.h"
2  #include <iostream>
3  using namespace std;
4
5  // Inisialisasi stack (membuat stack kosong)
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 stack (push)
21 void push(stack &S, infotype x) {
22     if (isFull(S)) {
23         cout << "Stack penuh, tidak bisa menambahkan elemen baru." << endl;
24     } else {
25         S.info[S.Top] = x;
26         S.Top++;
27     }
28 }
```

```

26     S.Top--;
27 }
28 }
29
30 // Menghapus elemen dari stack (pop)
31 int pop(stack &S) {
32     if (isEmpty(S)) {
33         cout << "Stack kosong, tidak ada elemen yang bisa dihapus." << endl;
34         return -1; // Mengembalikan -1 jika stack kosong
35     } else {
36         S.Top--; // Menurunkan Top
37         infotype x = S.info[S.Top]; // Simpan elemen yang dihapus
38         S.info[S.Top] = '\0'; // Hapus elemen dari stack
39         return x; // Mengembalikan elemen yang dihapus
40     }
41 }
42
43
44
45 // Mencetak semua elemen dalam stack
46 void printInfo(stack S) {
47     if (isEmpty(S)) {
48         cout << "Stack kosong." << endl;
49     } else {
50         cout << "Isi stack: ";
51         for (int i = 0; i < S.Top; i++) {
52             for (int j = 0; j < S.Top; j++) {
53                 cout << S.info[i] << " ";
54             }
55             cout << endl;
56         }
57

```

Main.cpp

```

TP > G: main.cpp > ...
1  #include <iostream>
2  #include "stack.h"
3  using namespace std;
4
5  void printStackMessage(stack &S, const char* message);
6
7  int main() {
8      stack S;
9      createStack(S);
10
11     int nimLastDigit;
12     cout << "Masukkan digit terakhir NIM Anda: ";
13     cin >> nimLastDigit;
14     nimLastDigit %= 4;
15
16     // Menentukan kata berdasarkan digit terakhir NIM MOD 4
17     switch(nimLastDigit) {
18         case 0: // Digit terakhir NIM MOD 4 sisa 0
19             // Masukkan "I F L A B J A Y A"
20             push(S, 'I');
21             push(S, 'F');
22             push(S, 'L');
23             push(S, 'A');
24             push(S, 'B');
25             push(S, 'J');
26             push(S, 'A');
27             push(S, 'Y');
28             push(S, 'A');
29             printStackMessage(S, "Isi stack awal: ");
30
31             // Pop untuk meninggalkan kata "J A Y A"
32             for (int i = 0; i < 5; i++) pop(S);
33             printStackMessage(S, "Isi stack sesudah pop: ");
34             break;
35

```

```

36     case 1: // Digit terakhir NIM MOD 4 sisa 1
37         // Masukkan "H A L O B A N D U N G"
38         push(S, 'H');
39         push(S, 'A');
40         push(S, 'L');
41         push(S, 'O');
42         push(S, 'B');
43         push(S, 'A');
44         push(S, 'N');
45         push(S, 'D');
46         push(S, 'U');
47         push(S, 'N');
48         push(S, 'G');
49         printStackMessage(S, "Isi stack awal: ");
50
51         // Pop untuk meninggalkan kata "B A N D U N G"
52         for (int i = 0; i < 4; i++) pop(S);
53         printStackMessage(S, "Isi stack sesudah pop: ");
54         break;
55
56     case 2: // Digit terakhir NIM MOD 4 sisa 2
57         // Masukkan "P E R C A Y A D I R I"
58         push(S, 'P');
59         push(S, 'E');
60         push(S, 'R');
61         push(S, 'C');
62         push(S, 'A');
63         push(S, 'Y');
64         push(S, 'A');
65         push(S, 'D');
66         push(S, 'I');
67         push(S, 'R');
68         push(S, 'I');
69         printStackMessage(S, "Isi stack awal: ");
70

```

```

71         // Pop untuk meninggalkan kata "D I R I"
72         for (int i = 0; i < 7; i++) pop(S); // Perbaiki menjadi 7 kali
73         printStackMessage(S, "Isi stack sesudah pop: ");
74         break;
75
76     case 3: // Digit terakhir NIM MOD 4 sisa 3
77         // Masukkan "S T R U K T U R D A T A"
78         push(S, 'S');
79         push(S, 'T');
80         push(S, 'R');
81         push(S, 'U');
82         push(S, 'K');
83         push(S, 'T');
84         push(S, 'U');
85         push(S, 'R');
86         push(S, 'D');
87         push(S, 'A');
88         push(S, 'T');
89         push(S, 'A');
90         printStackMessage(S, "Isi stack awal: ");
91
92         // Pop untuk meninggalkan kata "D A T A"
93         for (int i = 0; i < 8; i++) pop(S); // Perbaiki menjadi 8 kali
94         printStackMessage(S, "Isi stack sesudah pop: ");
95         break;
96
97     default:
98         cout << "Digit terakhir NIM tidak valid." << endl;
99         break;
100 }
101
102 return 0;
103 }

```

```

105 // Fungsi untuk mencetak pesan dan isi stack
106 void printStackMessage(stack &S, const char* message) {
107     cout << message;
108     printInfo(S);
109 }
110

```

Output:

```

Masukkan digit terakhir NIM Anda: 2
Isi stack awal: Isi stack: P E R C A Y A D I R I
Isi stack sesudah pop: Isi stack: P E R C
PS D:\Praktikum_07\TP>

```

Penjelasan:

Stack adalah struktur data yang mengikuti prinsip Last In First Out (LIFO), di mana elemen terakhir yang dimasukkan adalah elemen pertama yang dikeluarkan. Program ini dimulai dengan mendefinisikan tipe data infotype sebagai char dan mendeklarasikan struktur stack yang memiliki array info untuk menyimpan elemen dan variabel Top untuk menunjukkan indeks elemen teratas dalam stack.

Fungsi `createStack` digunakan untuk menginisialisasi stack dengan mengatur `Top` ke 0, menandakan bahwa stack kosong. Fungsi `isEmpty` dan `isFull` digunakan untuk memeriksa apakah stack kosong atau penuh, masing-masing. Fungsi `push` menambahkan elemen ke stack jika stack tidak penuh, sedangkan fungsi `pop` menghapus elemen teratas dari stack jika stack tidak kosong, dan mengembalikan nilai elemen yang dihapus. Fungsi `printInfo` mencetak semua elemen dalam stack.

Di dalam fungsi `main`, program meminta pengguna untuk memasukkan digit terakhir dari NIM mereka, yang kemudian digunakan untuk menentukan kata yang akan dimasukkan ke dalam stack berdasarkan hasil modulus 4 dari digit tersebut. Setiap kasus dalam `switch statement` mengatur kata yang berbeda untuk dimasukkan ke dalam stack. Setelah memasukkan kata, program melakukan operasi `pop` untuk menghapus sejumlah elemen dari stack, meninggalkan kata tertentu. Fungsi `printStackMessage` digunakan untuk mencetak pesan dan isi stack pada setiap langkah, memberikan umpan balik kepada pengguna tentang keadaan stack setelah setiap operasi. Program ini menunjukkan bagaimana stack dapat digunakan untuk menyimpan dan mengelola data dengan cara yang terstruktur.