

LAPORAN PRAKTIKUM
PERTEMUAN 7
STACK



Nama :

Ardhian Dwi Saputra (2311104040)

Dosen :

YUDHA ISLAMI SULISTYA

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

1. Stack.h

```
D: > SEMS 3 > Cpp std > Modul7 > TP > C 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 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
```

2. Stack.cpp

```
D: > SEMS 3 > Cpp std > Modul7 > TP > C Stack.cpp > ...
1  #include "stack.h"
2  #include <iostream>
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     } else {
22         cout << "Stack penuh, tidak bisa menambahkan elemen." << endl;
23     }
24 }
25
26 infotype pop(stack &S) {
27     if (!isEmpty(S)) {
28         infotype x = S.info[S.Top];
29         S.Top--;
30         return x;
31     } else {
32         cout << "Stack kosong, tidak ada elemen yang bisa diambil." << endl;
33         return '\0';
34     }
35 }
36
37 void printInfo(stack S) {
38     if (isEmpty(S)) {
39         cout << "Stack kosong." << endl;
40     } else {
41         for (int i = S.Top; i > 0; i--) {
42             cout << S.info[i] << " ";
43         }
44         cout << endl;
45     }
46 }
```

3. Main.cpp

```
D:\> SEMS 3 > Cpp std > Modul7 > TP > main.cpp > main()
1  #include "stack.h"
2  #include <iostream>
3  using namespace std;
4
5  int main() {
6      stack S;
7      createStack(S);
8
9      char frasaAwal[] = {'I', 'F', 'L', 'A', 'B', 'J', 'A', 'Y', 'A'};
10     char frasaPop[] = {'J', 'A', 'Y', 'A'};
11
12     for (char c : frasaAwal) {
13         push(S, c);
14     }
15
16     cout << "\nIsi stack awal: ";
17     printInfo(S);
18
19     cout << "Isi stack setelah pop: ";
20     for (char c : frasaPop) {
21         pop(S);
22     }
23     printInfo(S);
24
25     return 0;
26 }
```

Output:

```
Isi stack awal: A Y A J B A L F I
Isi stack setelah pop: B A L F I
PS D:\SEMS 3\Cpp std\modul7\TP>
```

Penjelasan code:

1. Stack.h

- #ifndef STACK_H dan #define STACK_H: Menghindari multiple inclusion (double include) dari file header.
- typedef char infotype;: Mendefinisikan tipe data infotype sebagai char.
- struct stack: Struktur stack memiliki array info berukuran 15 dan integer Top untuk melacak posisi teratas stack.
 - Fungsi-fungsi yang dideklarasikan:
 - createStack(stack &S): Inisialisasi stack kosong.
 - IsEmpty(stack S): Mengecek apakah stack kosong.
 - isFull(stack S): Mengecek apakah stack penuh.
 - push(stack &S, infotype x): Menambahkan elemen ke dalam stack.
 - pop(stack &S): Mengambil elemen dari stack.
 - printInfo(stack S): Menampilkan semua elemen dalam stack.

2. Stack.cpp

- createStack(stack &S): Mengatur Top ke 0 untuk menunjukkan stack

kosong.

- isEmpty(stack S): Mengembalikan true jika Top == 0, artinya stack kosong.
- isFull(stack S): Mengembalikan true jika Top == 15, artinya stack penuh.
- push(stack &S, infotype x): Menambahkan elemen x ke dalam stack jika stack belum penuh. Jika penuh, menampilkan pesan "Stack penuh".
- pop(stack &S): Mengeluarkan elemen di posisi Top dan mengembalikannya. Jika stack kosong, menampilkan pesan "Stack kosong" dan mengembalikan '\0'.
- printInfo(stack S): Menampilkan semua elemen dari Top ke 1. Jika stack kosong, menampilkan pesan "Stack kosong".

3. Main.cpp

- Deklarasi dan Inisialisasi Stack: Membuat stack S dan memanggil createStack(S) untuk menginisialisasi.
- Mengisi Stack: Memasukkan karakter-karakter dari frasaAwal ('I', 'F', 'L', 'A', 'B', 'J', 'A', 'Y', 'A') ke dalam stack menggunakan push.
- Menampilkan Isi Stack Awal: Menampilkan isi stack setelah semua karakter dari frasaAwal dimasukkan.
- Mengeluarkan Elemen Stack dengan pop: Mengeluarkan elemen sebanyak karakter di frasaPop ('J', 'A', 'Y', 'A').
- Menampilkan Isi Stack Setelah Pop: Menampilkan sisa elemen dalam stack setelah beberapa elemen dikeluarkan.

Unguided

Soal 1

```

D: > SEMS 3 > Cpp std > Modul7 > Unguided_01.cpp > isPalindrome(string)
1  #include <iostream>
2  #include <stack>
3  #include <cctype>
4  using namespace std;
5
6  bool isPalindrome(string str) {
7      stack<char> s;
8
9      for (char c : str) {
10         if (isalpha(c)) {
11             s.push(tolower(c));
12         }
13     }
14
15     for (char c : str) {
16         if (isalpha(c)) {
17             if (tolower(c) != s.top()) {
18                 return false;
19             }
20             s.pop();
21         }
22     }
23     return true;
24 }
25
26 int main() {
27     string kalimat;
28     cout << "Masukkan kalimat: ";
29     getline(cin, kalimat);
30
31     if (isPalindrome(kalimat)) {
32         cout << "Kalimat adalah palindrom." << endl;
33     } else {
34         cout << "Kalimat bukan palindrom." << endl;
35     }
36
37     return 0;
38 }

```

Penjelasan code:

- <iostream>: Untuk input dan output.
- <stack>: Menggunakan stack untuk membalikkan karakter.
- <cctype>: Menggunakan fungsi isalpha untuk mengecek huruf dan tolower untuk mengubah karakter menjadi huruf kecil.

Fungsi ini mengecek apakah str adalah palindrom.

- Penggunaan Stack:
- Loop Pertama: Mengisi stack s dengan setiap huruf dalam str, setelah mengonversi ke huruf kecil (tolower(c)). Hanya huruf (isalpha(c)) yang dimasukkan ke dalam stack, sehingga spasi dan tanda baca diabaikan.
- Loop Kedua: Membandingkan setiap huruf dalam str dengan karakter teratas

(s.top()) di stack (dimulai dari awal hingga akhir). Setiap kali ada kecocokan, karakter di str dan s di-pop dari stack. Jika ada yang tidak cocok, fungsi mengembalikan false.

- **Return Value:** Jika seluruh karakter cocok, fungsi mengembalikan true, menandakan str adalah palindrom.
- **Input:** Mengambil kalimat dari pengguna menggunakan getline.
- **Memanggil isPalindrome:** Mengecek apakah kalimat adalah palindrom.

Output :

```
Masukkan kalimat: ini
Kalimat adalah palindrom.
PS D:\SEMS 3\Cpp std\modul5>
```

Soal 2

```
D: > SEMS 3 > Cpp std > Modul7 > Unguided_02.cpp > ...
1  #include <iostream>
2  #include <stack>
3  #include <sstream>
4  using namespace std;
5
6  void kalimatMundur(string kalimat) {
7      stack<string> s;
8      stringstream ss(kalimat);
9      string kata;
10
11      while (ss >> kata){
12          s.push(kata);
13      }
14
15      cout << "Kalimat Kebalik: ";
16      while (!s.empty()) {
17          cout << s.top() << " ";
18          s.pop();
19      }
20  }
21
22  int main(){
23
24      string kalimat;
25
26      cout << "\nMasukan 3 Kalimat: " ;
27      getline(cin, kalimat);
28      kalimatMundur(kalimat);
29
30      return 0 ;
31  }
```

Penjelasan code:

- `<iostream>`: Untuk input dan output.
 - `<stack>`: Menggunakan stack untuk menyimpan kata-kata dari kalimat yang dimasukkan.
 - `<sstream>`: Menggunakan stringstream untuk memisahkan kata-kata dalam kalimat berdasarkan spasi.
 - `<iostream>`: Untuk input dan output.
 - `<stack>`: Menggunakan stack untuk menyimpan kata-kata dari kalimat yang dimasukkan.
 - `<sstream>`: Menggunakan stringstream untuk memisahkan kata-kata dalam kalimat berdasarkan spasi.
- Fungsi ini menerima parameter kalimat dan mencetak kalimat tersebut dalam urutan terbalik.

- Deklarasi Stack dan Stringstream:
- `stack<string> s`: Stack s untuk menyimpan setiap kata dalam urutan masuknya.
- `stringstream ss(kalimat)`: stringstream digunakan untuk membaca kata-kata dari kalimat berdasarkan spasi.
- Loop Pertama:
- `while (ss >> kata)`: Setiap kali menemukan kata dalam kalimat, kata tersebut disimpan dalam variabel kata dan kemudian dimasukkan (push) ke dalam stack s.
- Hasilnya, setiap kata dari kalimat tersimpan dalam stack dari urutan pertama hingga terakhir.
- Menampilkan Kalimat Terbalik:
- `cout << "Kalimat Kebalik: ";` Mencetak teks "Kalimat Kebalik:" sebagai pengantar.
- `while (!s.empty())`: Selama stack tidak kosong, ambil dan cetak (`s.top()`) kata paling atas, lalu keluarkan kata tersebut dari stack (`s.pop()`).
- Karena stack adalah struktur LIFO (Last In, First Out), proses ini akan mencetak kata-kata dalam urutan terbalik dari kalimat asli.
- Input Kalimat: Mengambil input dari pengguna menggunakan `getline(cin, kalimat);`, sehingga seluruh kalimat dapat dimasukkan termasuk spasi.
- Memanggil Fungsi kalimatMundur: Mengirimkan kalimat yang diinput ke fungsi kalimatMundur untuk diproses.
- Output: Fungsi kalimatMundur akan menampilkan kalimat dengan urutan kata yang dibalik.

Output:

```
Masukan 3 Kalimat: dia suka kamu
Kalimat Kebalik: kamu suka dia
PS D:\SEMS 3\Cpp std\modul5> 
```