

LAPORAN PRAKTIKUM

MODUL 7

STACK



Disusun Oleh:

Rizaldy Aulia Rachman (2311104051)

S1SE-07-02

Dosen :

Wahyu Andi Saputra, S.Pd., M.Eng

PROGRAM STUDI S1 SOFTWARE ENGINEERING

FAKULTAS INFORMATIKA

TELKOM UNIVERSITY PURWOKERTO

2024

I. TUJUAN

1. Mampu memahami konsep stack pada struktur data dan algoritma.
2. Mampu mengimplementasikan operasi-operasi pada stack.
3. Mampu memecahkan permasalahan dengan solusi stack.

II. LANDASAN TEORI

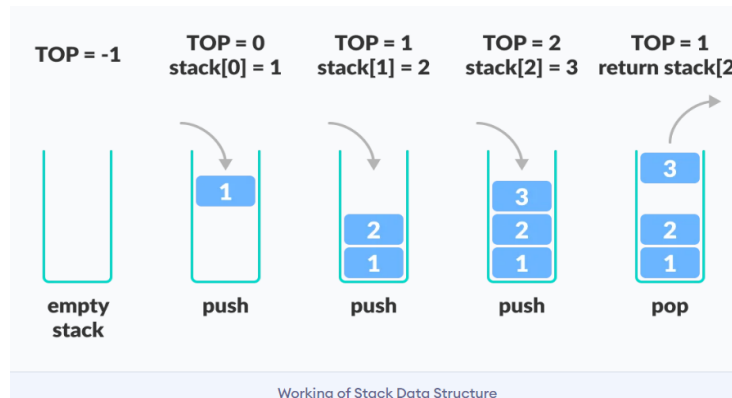
2.1 Stack

Sebuah stack atau tumpukan merupakan struktur data yang berfungsi untuk menyimpan dan mengelola kumpulan data dengan prinsip Last-In, First-Out (LIFO). Analogi yang sering digunakan adalah tumpukan piring di kafetaria, di mana piring terakhir yang ditambahkan akan menjadi yang pertama diambil.

Dalam implementasinya, stack dapat direpresentasikan sebagai struktur data terurut yang memiliki dua operasi utama: push dan pop. Operasi push digunakan untuk menambahkan elemen baru ke dalam stack, sementara operasi pop digunakan untuk menghapus elemen teratas dari stack.

Prinsip LIFO yang menjadi dasar stack membuatnya sangat bermanfaat dalam berbagai aplikasi, termasuk manajemen memori komputer, evaluasi ekspresi aritmatika, dan manajemen panggilan fungsi dalam pemrograman. Sebagai contoh, dalam manajemen memori, stack digunakan untuk menyimpan alamat-alamat memori yang dialokasikan untuk variabel dan fungsi.

Dengan prinsip LIFO ini, stack memungkinkan akses data dengan efisiensi, di mana elemen yang terakhir dimasukkan akan menjadi yang pertama diambil. Hal ini menjadikannya salah satu struktur data yang sangat penting dalam pengembangan perangkat lunak dan pemrograman komputer.



2.2 Operasi-Operasi dalam Stack

Operasi pada stack melibatkan beberapa fungsi dasar yang dapat dilakukan pada struktur data ini. Berikut adalah beberapa operasi umum pada stack:

- a. Push (Masukkan): Menambahkan elemen ke dalam tumpukan pada posisi paling atas atau ujung.
- b. Pop (Keluarkan): Menghapus elemen dari posisi paling atas atau ujung tumpukan.
- c. Top (Atas): Mendapatkan nilai atau melihat elemen teratas pada tumpukan tanpa menghapusnya.
- d. IsEmpty (Kosong): Memeriksa apakah tumpukan kosong atau tidak.
- e. IsFull (Penuh): Memeriksa apakah tumpukan penuh atau tidak (terutama pada implementasi tumpukan dengan kapasitas terbatas).
- f. Size (Ukuran): Mengembalikan jumlah elemen yang ada dalam tumpukan.
- g. Peek (Lihat): Melihat nilai atau elemen pada posisi tertentu dalam tumpukan tanpa menghapusnya.
- h. Clear (Hapus Semua): Mengosongkan atau menghapus semua elemen dari tumpukan.
- i. Search (Cari): Mencari keberadaan elemen tertentu dalam tumpukan.

III. GUIDED

1. Guided1

Code:

```

1  #include <iostream>
2  #define MAX 100
3
4  using namespace std;
5
6  class Stack {
7  private:
8      int top;
9      int arr[MAX];
10
11 public:
12     Stack() { top = -1; }
13
14     bool isFull() { return top == MAX - 1; }
15     bool isEmpty() { return top == -1; }
16
17     void push(int x) {
18         if (isFull()) {
19             cout << "Stack Overflow\n";
20             return;
21         }
22         arr[++top] = x;
23     }
24
25     void pop() {
26         if (isEmpty()) {
27             cout << "Stack Underflow\n";
28             return;
29         }
30         top--;
31     }
32
33     int peek() {
34         if (!isEmpty()) {
35             return arr[top];
36         }
37         cout << "Stack is empty\n";
38         return -1;
39     }
40
41     void display() {
42         if (isEmpty()) {
43             cout << "Stack is empty\n";
44             return;
45         }
46         for (int i = top; i >= 0; i--) {
47             cout << arr[i] << " ";
48         }
49         cout << "\n";
50     }
51 };
52
53 int main() {
54     Stack s;
55     s.push(10);
56     s.push(20);
57     s.push(30);
58
59     cout << "Stack elements: ";
60     s.display();
61
62     cout << "Top element: " << s.peek() << "\n";
63
64     s.pop();
65     s.pop();
66     cout << "After popping, stack elements: ";
67     s.display();
68
69     return 0;
70 }

```

Output:

```
Stack elements: 30 20 10  
Top element: 30  
After popping, stack elements: 10  
PS C:\Praktikum Struktur data\pertemuan7>
```

2. Guided2

Code:

```

1  #include <iostream>
2
3  using namespace std;
4
5  class Node {
6  public:
7      int data;
8      Node* next;
9      Node(int value) {
10         data = value;
11         next = nullptr;
12     }
13 };
14
15 class Stack {
16 private:
17     Node* top;
18
19 public:
20     Stack() { top = nullptr; }
21     bool isEmpty() { return top == nullptr; }
22
23     void push(int x) {
24         Node* newNode = new Node(x);
25         newNode->next = top;
26         top = newNode;
27     }
28
29     void pop() {
30         if (isEmpty()) {
31             cout << "Stack Underflow\n";
32             return;
33         }
34         Node* temp = top;
35         top = top->next;
36         delete temp;
37     }
38
39     int peek() {
40         if (!isEmpty()) {
41             return top->data;
42         }
43         cout << "Stack is empty\n";
44         return -1; // Return a sentinel value
45     }
46
47     void display() {
48         if (isEmpty()) {
49             cout << "Stack is empty\n";
50             return;
51         }
52         Node* current = top;
53         while (current) {
54             cout << current->data << " ";
55             current = current->next;
56         }
57         cout << "\n";
58     }
59 };
60
61 int main() {
62     Stack s;
63     s.push(10);
64     s.push(20);
65     s.push(30);
66
67     cout << "Stack elements: ";
68     s.display();
69
70     cout << "Top element: " << s.peek() << "\n";
71
72     s.pop();
73     s.pop();
74     cout << "After popping, stack elements: ";
75     s.display();
76
77     return 0;
78 }

```

Output:

```
Stack elements: 30 20 10
Top element: 30
After popping, stack elements: 10
PS C:\Praktikum Struktur data\pertemuan7>
```

IV. 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.

Jawaban:

Code:

```

1  #include <iostream>
2  #include <stack>
3  #include <algorithm>
4  #include <cctype>
5
6  using namespace std;
7
8  bool isPalindrome(string sentence) {
9      stack<char> charStack;
10     string cleanSentence = "";
11
12     for (char c : sentence) {
13         if (isalnum(c)) {
14             cleanSentence += tolower(c);
15         }
16     }
17
18     for (char c : cleanSentence) {
19         charStack.push(c);
20     }
21
22     string reversedSentence = "";
23     while (!charStack.empty()) {
24         reversedSentence += charStack.top();
25         charStack.pop();
26     }
27
28     return cleanSentence == reversedSentence;
29 }
30
31 int main() {
32     string sentence;
33     cout << "Masukkan kalimat: ";
34     getline(cin, sentence);
35
36     if (isPalindrome(sentence)) {
37         cout << "Kalimat tersebut adalah palindrom." << endl;
38     } else {
39         cout << "Kalimat tersebut bukan palindrom." << endl;
40     }
41
42     return 0;
43 }
44

```

Output:

```

Masukkan kalimat: ini
Kalimat tersebut adalah palindrom.
PS C:\Praktikum Struktur data\pertemuan7>

```

```

Masukkan kalimat: telkom
Kalimat tersebut bukan palindrom.
PS C:\Praktikum Struktur data\pertemuan7>

```

Penjelasan cara kerja program:

1. **Input diproses** untuk menghapus spasi, simbol, dan mengubah huruf besar menjadi kecil.
2. **Karakter dimasukkan ke stack**, lalu diambil kembali untuk membentuk string terbalik.
3. **String asli dibandingkan** dengan string terbalik untuk memutuskan apakah palindrom.

2. Buatlah program untuk melakukan pembalikan terhadap kalimat menggunakan stack dengan minimal 3 kata. Jelaskan output program dan source codenya beserta operasi/fungsi yang dibuat?

Jawaban:

Code:

```

1  #include <iostream>
2  #include <stack>
3  #include <sstream>
4  #include <algorithm>
5
6  using namespace std;
7
8  string reverseWord(string word) {
9      stack<char> charStack;
10
11     for (char c : word) {
12         charStack.push(c);
13     }
14
15     string reversedWord = "";
16     while (!charStack.empty()) {
17         reversedWord += charStack.top();
18         charStack.pop();
19     }
20
21     return reversedWord;
22 }
23
24 string reverseSentenceWithWords(string sentence) {
25     stack<string> wordStack;
26     stringstream ss(sentence);
27     string word;
28
29     while (ss >> word) {
30         wordStack.push(reverseWord(word));
31     }
32
33     string reversedSentence = "";
34     while (!wordStack.empty()) {
35         reversedSentence += wordStack.top();
36         wordStack.pop();
37         if (!wordStack.empty()) {
38             reversedSentence += " ";
39         }
40     }
41
42     return reversedSentence;
43 }
44
45 int main() {
46     string sentence;
47     cout << "Masukkan kalimat minimal 3 kata: ";
48     getline(cin, sentence);
49
50     stringstream ss(sentence);
51     int wordCount = 0;
52     string temp;
53     while (ss >> temp) {
54         wordCount++;
55     }
56
57     if (wordCount < 3) {
58         cout << "Kalimat harus minimal 3 kata." << endl;
59     } else {
60         cout << "Kalimat yang dibalik (termasuk kata-katanya): " << reverseSentenceWithWords(sentence) << endl;
61     }
62
63     return 0;
64 }
65

```

Output:

```

Masukkan kalimat minimal 3 kata: Telkom University Purwokerto
Kalimat yang dibalik (termasuk kata-katanya): otrekowruP ytisrevinU mokleT
PS C:\Praktikum Struktur data\pertemuan7>

```

Penjelasan Program:

1. Fungsi reverseWord:

- Menggunakan stack untuk membalikkan karakter dalam satu kata.
 - Karakter dari kata dimasukkan ke stack, lalu dikeluarkan untuk membentuk kata terbalik.
2. Fungsi reverseSentenceWithWords:
- Membagi kalimat menjadi kata-kata menggunakan stringstream.
 - Setiap kata dibalik menggunakan fungsi reverseWord, lalu dimasukkan ke stack.
 - Urutan kata dibalik dengan mengambilnya dari stack satu per satu.
3. Proses di main:
- Memvalidasi jumlah kata di kalimat (minimal 3 kata).
 - Menampilkan hasil berupa kalimat dengan kata-kata yang terbalik.