

Tugas Pendahuluan Modul 7

STRUKTUR DATA - Ganjil 2024/2025

Stack

Ketentuan Tugas Pendahuluan

1. Tugas Pendahuluan dikerjakan secara **Individu**.
2. TP ini bersifat **WAJIB**, tidak mengerjakan = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
3. Hanya **MENGUMPULKAN** tetapi **TIDAK MENGERJAKAN** = **PENGURANGAN POIN JURNAL / TES ASESMEN**.
4. Deadline pengumpulan TP Modul 2 adalah Senin, 30 September 2024 pukul 07.30 WIB.
5. **TIDAK ADA TOLERANSI KETERLAMBATAN, TERLAMBAT ATAU TIDAK MENGUMPULKAN TP MAKA DIANGGAP TIDAK MENGERJAKAN**.
6. **DILARANG PLAGIAT (PLAGIAT = E)**.
7. Kerjakan TP dengan jelas agar dapat dimengerti.
8. Codingan diupload di Github dan upload Laporan di Lab menggunakan format **PDF** dengan ketentuan: **TP_MOD_[XX]_NIM_NAMA.pdf**

CP (WA):

- Andini (082243700965)
- Imelda (082135374187)

SELAMAT MENGERJAKAN^^

LAPORAN PRAKTIKUM
PERTEMUAN 7
STACK



Nama :

Tsaqif Hisyam Saputra (2311104024)

Dosen :

Yudha Islami Sulistya

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

A. Tujuan

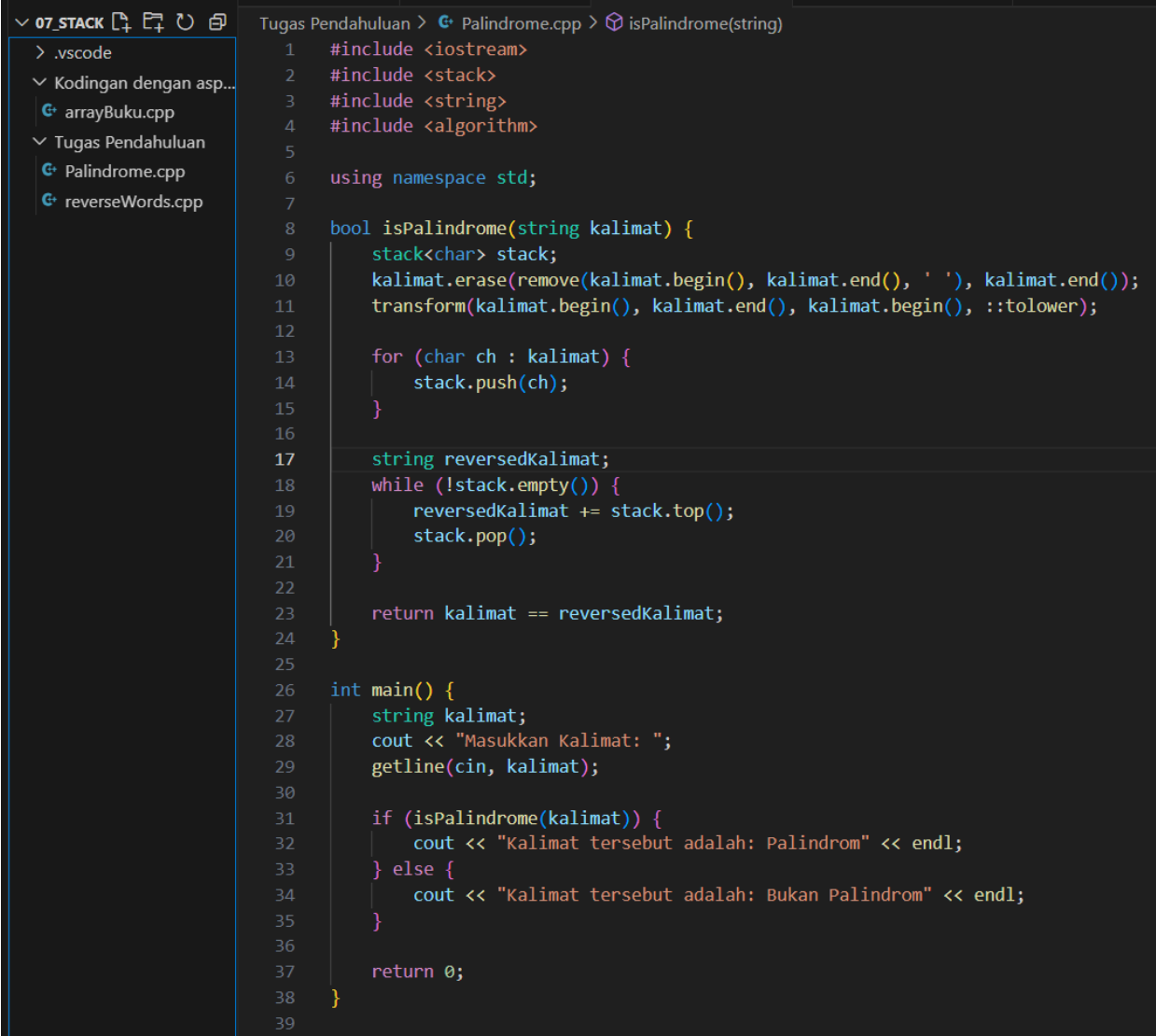
- Memahami konsep *stack*.
- Mengimplementasikan *stack* dengan menggunakan *pointer* dan tabel.
- Memahami konsep *queue*.
- Mengimplementasikan *queue* dengan menggunakan *pointer* dan tabel

B. Tools

Visual Studio Code dengan C++ Extensions Pack & Codeblocks

Tugas Pendahuluan

Palindorme



```
1  #include <iostream>
2  #include <stack>
3  #include <string>
4  #include <algorithm>
5
6  using namespace std;
7
8  bool isPalindrome(string kalimat) {
9      stack<char> stack;
10     kalimat.erase(remove(kalimat.begin(), kalimat.end(), ' '), kalimat.end());
11     transform(kalimat.begin(), kalimat.end(), kalimat.begin(), ::tolower);
12
13     for (char ch : kalimat) {
14         stack.push(ch);
15     }
16
17     string reversedKalimat;
18     while (!stack.empty()) {
19         reversedKalimat += stack.top();
20         stack.pop();
21     }
22
23     return kalimat == reversedKalimat;
24 }
25
26 int main() {
27     string kalimat;
28     cout << "Masukkan Kalimat: ";
29     getline(cin, kalimat);
30
31     if (isPalindrome(kalimat)) {
32         cout << "Kalimat tersebut adalah: Palindrom" << endl;
33     } else {
34         cout << "Kalimat tersebut adalah: Bukan Palindrom" << endl;
35     }
36
37     return 0;
38 }
39
```

Pada Program Palindorme di atas menggunakan fungsi **stack** untuk membalik urutan karakter. Setiap karakter (**ch**) dalam kalimat dimasukkan ke **stack**, lalu dikeluarkan dalam urutan terbalik untuk membentuk **reversedKalimat**. Jika **reversedKalimat** sama dengan kalimat asli yang diproses, maka kalimat tersebut adalah palindrom, dan jika kalimat yang diproses tidak bisa dibalik lagi saat kembali maka kalimat tersebut bukan palindrom.

OUTPUT:

```
● * Executing task: C:/Windows/System32/cmd.exe /d /c .\build\Debu

Masukkan Kalimat: Race Car
Kalimat tersebut adalah: Palindrom
* Terminal will be reused by tasks, press any key to close it.

● * Executing task: C:/Windows/System32/cmd.exe /d /c .\build\Debu

Masukkan Kalimat: Verstappen
Kalimat tersebut adalah: Bukan Palindrom
* Terminal will be reused by tasks, press any key to close it.
```

Reverse Words

```
Tugas Pendahuluan > reverseWords.cpp > reverseWords(string)
1  #include <iostream>
2  #include <stack>
3  #include <sstream>
4  #include <string>
5
6  using namespace std;
7
8  string reverseWords(string sentence) {
9      stack<char> stack;
10     stringstream ss(sentence);
11     string word, reversedSentence;
12
13     while (ss >> word) {
14         for (char ch : word) {
15             stack.push(ch);
16         }
17
18         while (!stack.empty()) {
19             reversedSentence += stack.top();
20             stack.pop();
21         }
22         reversedSentence += " ";
23     }
24
25     if (!reversedSentence.empty()) {
26         reversedSentence.pop_back();
27     }
28
29     return reversedSentence;
30 }
31
32 int main() {
33     string kalimat;
34     cout << "Masukkan Kalimat (minimal 3 kata): ";
35     getline(cin, kalimat);
36
37     string hasil = reverseWords(kalimat);
38     cout << "Hasil: " << hasil << endl;
39
40     return 0;
41 }
```

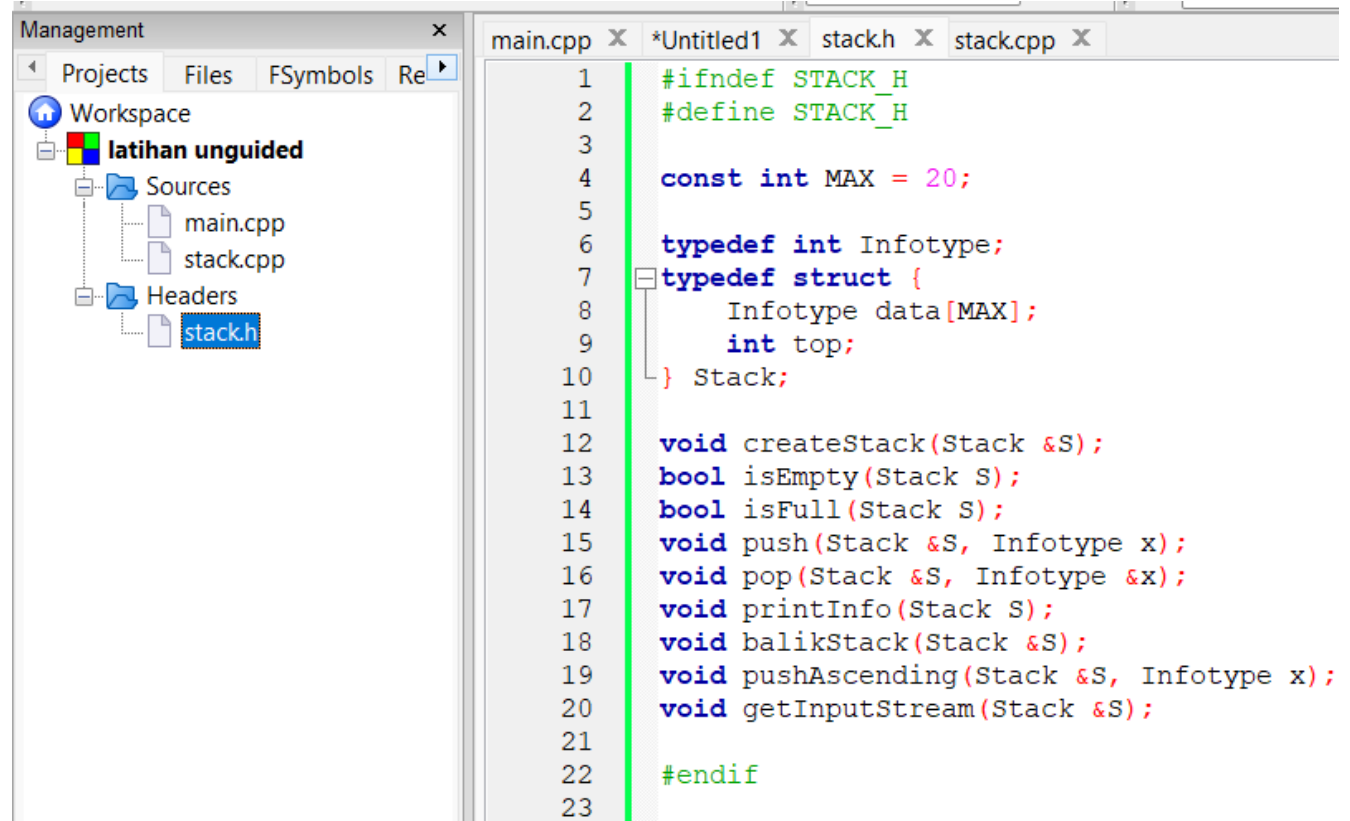
Lalu dalam pemrograman di `reverseWords`, `&word` digunakan pada `stringstream ss(sentence)`; untuk membaca setiap kata dalam kalimat **sentence** secara terpisah. Saat `ss >> word` dijalankan, **word** menyimpan kata berikutnya dalam kalimat, yang kemudian dibalik menggunakan `stack` dan ditambahkan ke **reversedSentence**.

OUTPUT:

```
● * Executing task: C:/Windows/System32/cmd.exe /d /c .\build\Debug  
  
Masukkan Kalimat (minimal 3 kata): menghitung dengan sigma  
Hasil: gnutihgnem nagned amgis  
* Terminal will be reused by tasks, press any key to close it.
```

Latihan Unguided

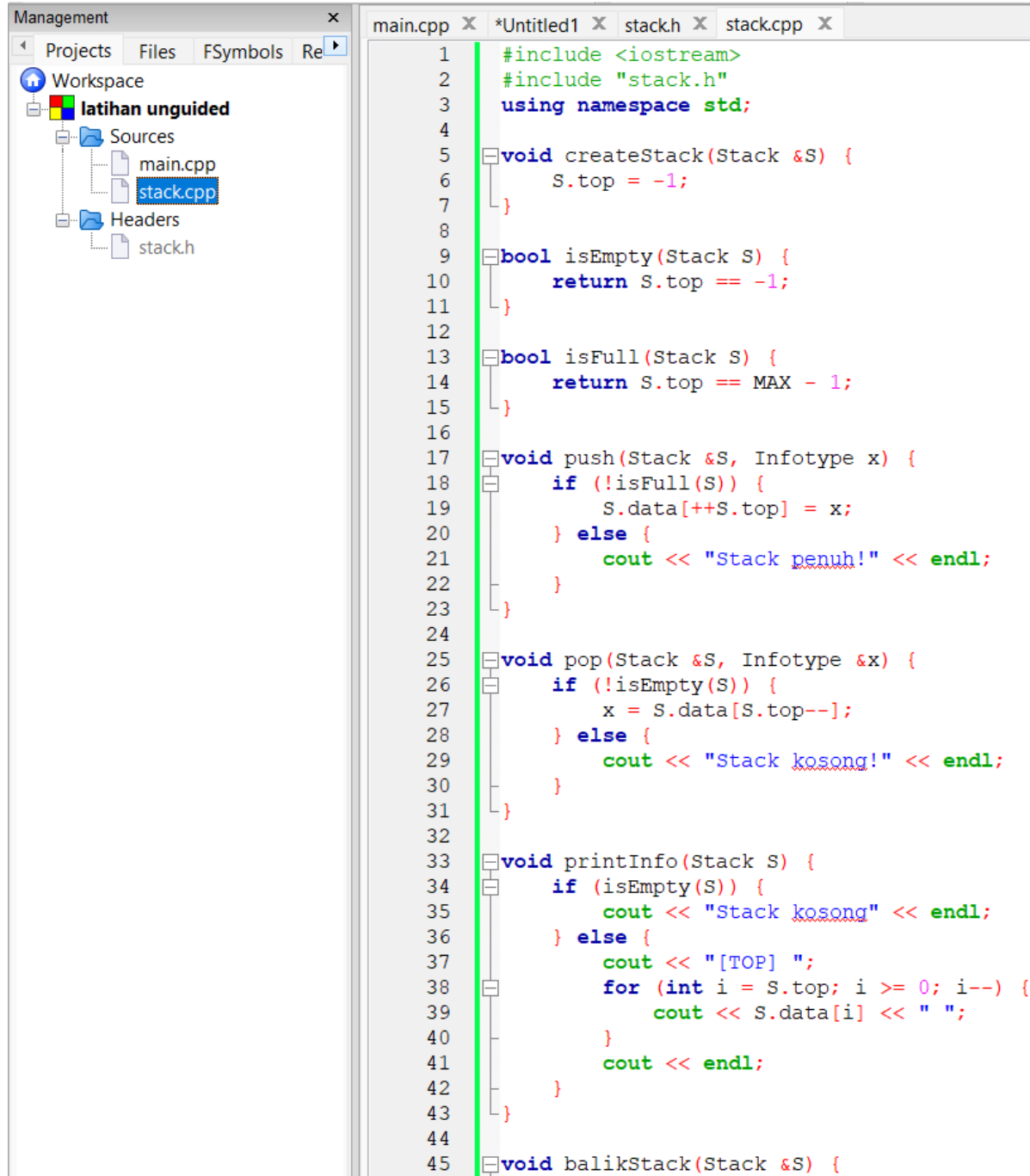
1. stack.h



```
1  #ifndef STACK_H
2  #define STACK_H
3
4  const int MAX = 20;
5
6  typedef int Infotype;
7  typedef struct {
8      Infotype data[MAX];
9      int top;
10 } Stack;
11
12 void createStack(Stack &S);
13 bool isEmpty(Stack S);
14 bool isFull(Stack S);
15 void push(Stack &S, Infotype x);
16 void pop(Stack &S, Infotype &x);
17 void printInfo(Stack S);
18 void balikStack(Stack &S);
19 void pushAscending(Stack &S, Infotype x);
20 void getInputStream(Stack &S);
21
22 #endif
23
```

Kodingan di **stack.h** adalah file header yang mendefinisikan struktur data **Stack** dan deklarasi fungsi-fungsi buat operasi stack, seperti **createStack**, **push**, **pop**, **printInfo**, **balikStack**, **pushAscending**, dan **getInputStream**. File ini memungkinkan pemisahan antara deklarasi dan implementasi fungsi, sehingga mempermudah pengelolaan dan penggunaan **Stack** dalam program utama.

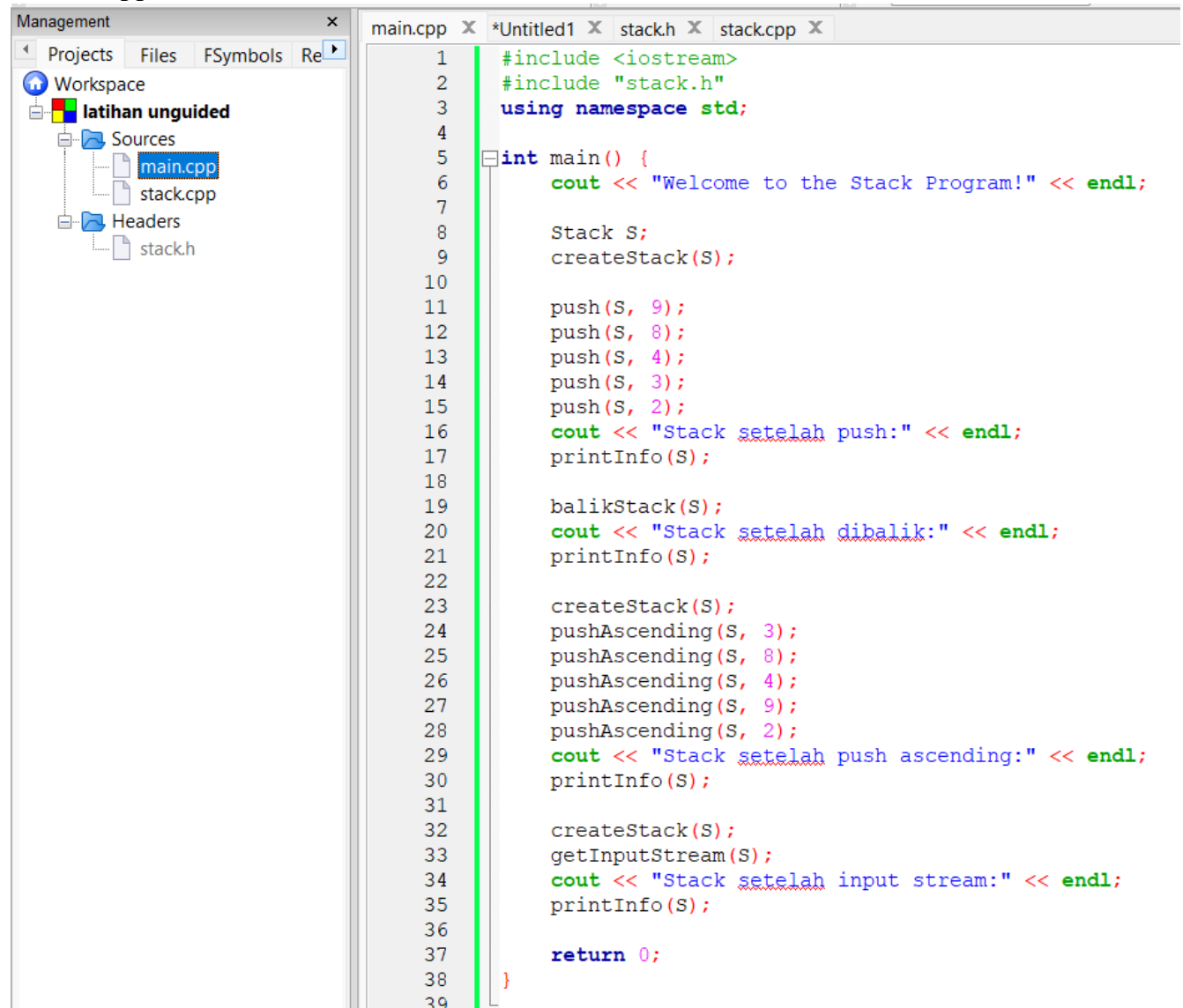
2. stack.cpp



```
1  #include <iostream>
2  #include "stack.h"
3  using namespace std;
4
5  void createStack(Stack &S) {
6      S.top = -1;
7  }
8
9  bool isEmpty(Stack S) {
10     return S.top == -1;
11 }
12
13 bool isFull(Stack S) {
14     return S.top == MAX - 1;
15 }
16
17 void push(Stack &S, Infotype x) {
18     if (!isFull(S)) {
19         S.data[++S.top] = x;
20     } else {
21         cout << "Stack penuh!" << endl;
22     }
23 }
24
25 void pop(Stack &S, Infotype &x) {
26     if (!isEmpty(S)) {
27         x = S.data[S.top--];
28     } else {
29         cout << "Stack kosong!" << endl;
30     }
31 }
32
33 void printInfo(Stack S) {
34     if (isEmpty(S)) {
35         cout << "Stack kosong" << endl;
36     } else {
37         cout << "[TOP] ";
38         for (int i = S.top; i >= 0; i--) {
39             cout << S.data[i] << " ";
40         }
41         cout << endl;
42     }
43 }
44
45 void balikStack(Stack &S) {
```

Dalam **stack.cpp** mengimplementasikan fungsi-fungsi yang dideklarasikan di **stack.h** untuk operasi **Stack**. Ini mencakup inisialisasi (**createStack**), penambahan (**push**), pengeluaran (**balikStack**). Penyisipan berurutan (**pushAscending**), dan membaca input stream (**getInputStream**). File ini memisahkan logika implementasi dari deklarasi, mempermudah pemeliharaan program.

3. main.cpp



```
1  #include <iostream>
2  #include "stack.h"
3  using namespace std;
4
5  int main() {
6      cout << "Welcome to the Stack Program!" << endl;
7
8      Stack S;
9      createStack(S);
10
11     push(S, 9);
12     push(S, 8);
13     push(S, 4);
14     push(S, 3);
15     push(S, 2);
16     cout << "Stack setelah push:" << endl;
17     printInfo(S);
18
19     balikStack(S);
20     cout << "Stack setelah dibalik:" << endl;
21     printInfo(S);
22
23     createStack(S);
24     pushAscending(S, 3);
25     pushAscending(S, 8);
26     pushAscending(S, 4);
27     pushAscending(S, 9);
28     pushAscending(S, 2);
29     cout << "Stack setelah push ascending:" << endl;
30     printInfo(S);
31
32     createStack(S);
33     getInputStream(S);
34     cout << "Stack setelah input stream:" << endl;
35     printInfo(S);
36
37     return 0;
38 }
39
```

Lalu **main.cpp** adalah file utama yang menguji fungsi-fungsi stack yang telah diimplementasikan. Dalam program ini, **createStack** digunakan untuk menginisialisasi stack, dan fungsi yang dibilang diatas dan **getInputStream** diuji untuk melihat hasilnya. **printInfo** akan menampilkan hasil dari setiap fungsi yang mempengaruhi data di stack.

OUTPUT:

```
Masukkan angka (akhiri dengan Enter): 13589  
Stack setelah input stream:  
[TOP] 9 8 5 3 1
```

```
Masukkan angka (akhiri dengan Enter): 98887631  
Stack setelah input stream:  
[TOP] 1 3 6 7 8 8 8 9
```

Kodingan di Kelas

```
1 #include <iostream>
2 using namespace std;
3
4 string arrayBuku[5];
5 int maksimal = 5, top = 0;
6
7 bool isFull()
8 {
9     return top == maksimal;
10 }
11
12 bool isEmpty()
13 {
14     return top == 0;
15 }
16
17 void pushArrayBuku(string data)
18 {
19     if (isFull())
20     {
21         cout << "Stack-nya penuh yow" << endl;
22     }
23     else
24     {
25         arrayBuku[top] = data;
26         top++;
27     }
28 }
29
30 void popArrayBuku()
31 {
32     if (isEmpty())
33     {
34         cout << "Gada data yang dihapus" << endl;
35     }
36     else
37     {
38         arrayBuku[top - 1] = ""; // Menghapus elemen terakhir
39         top--;
40     }
41 }
42
43 void peekArrayBuku(int posisi)
44 {
45     if (isEmpty())
46     {
47         cout << "Gada data yang bisa dilihat" << endl;
48     }
49     else
50     {
51         int index = top - posisi;
52         if (index >= 0 && index < top)
53         {
54             cout << "Posisi ke- " << posisi << " adalah " << arrayBuku[index] << endl;
55         }
56         else
57         {
58             cout << "Posisi diluar jangkauan" << endl;
59         }
60     }
61 }
62
63 int countStack()
64 {
65     return top;
66 }
67
68 void changeArrayBuku(int posisi, string data)
69 {
70     int index = top - posisi;
71     if (index >= 0 && index < top)
72     {
73         arrayBuku[index] = data;
74     }
75     else
76     {
77         cout << "Kelelahan posisinya" << endl;
78     }
79 }
80
81 void destroyArrayBuku()
82 {
83     for (int i = top - 1; i >= 0; i--)
84     {
85         arrayBuku[i] = "";
86     }
87     top = 0;
88 }
89
90 void cetakArrayBuku()
91 {
92     if (isEmpty())
93     {
94         cout << "Gada data yang bisa dicetak" << endl;
95     }
96     else
97     {
98         for (int i = top - 1; i >= 0; i--)
99         {
100             cout << arrayBuku[i] << endl;
101         }
102     }
103 }
104
105 int main()
106 {
107     pushArrayBuku("Kalkulus");
108     pushArrayBuku("Bahasa Indonesia");
109     pushArrayBuku("IPA");
110     pushArrayBuku("IPS");
111     pushArrayBuku("Kilat");
112
113     cetakArrayBuku();
114     cout << "Ya" << endl;
115
116     cout << "Stack-nya penuh?" << (isFull() ? "Ya" : "Tidak") << endl;
117     cout << "Stack-nya kosong?" << (isEmpty() ? "Ya" : "Tidak") << endl;
118
119     peekArrayBuku(2);
120     popArrayBuku();
121
122     cout << "Keseluruhan data = " << countStack() << endl;
123
124     changeArrayBuku(2, "Bahasa Jerman");
125     cetakArrayBuku();
126     cout << "Ya" << endl;
127
128     destroyArrayBuku();
129     cout << "Tepi setelah destroy : " << top << endl;
130     cetakArrayBuku();
131
132     return 0;
133 }
```

Semoga Selalu diberi kemudahan^^