

LAPORAN PRAKTIKUM
PERTEMUAN 7
STACK



Nama :

Salman Alfarisi (2311104036)

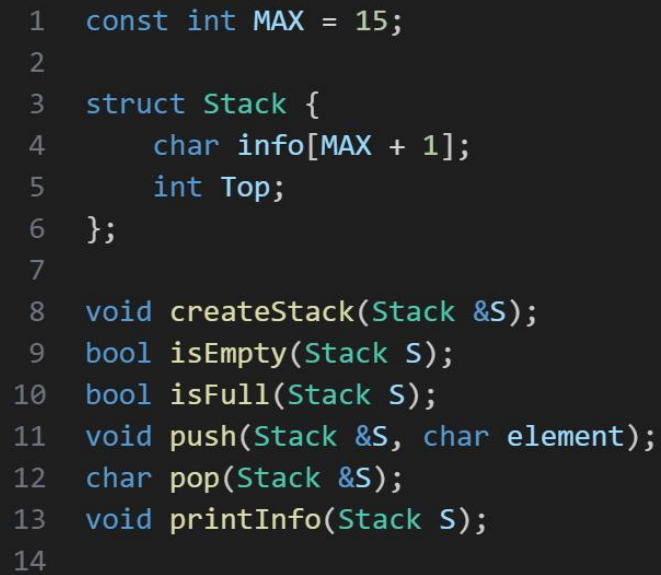
Dosen :

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

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

TUGAS PENDAHULUAN 1

CLASS stack.h



```
1  const int MAX = 15;
2
3  struct Stack {
4      char info[MAX + 1];
5      int Top;
6  };
7
8  void createStack(Stack &S);
9  bool isEmpty(Stack S);
10 bool isFull(Stack S);
11 void push(Stack &S, char element);
12 char pop(Stack &S);
13 void printInfo(Stack S);
14
```

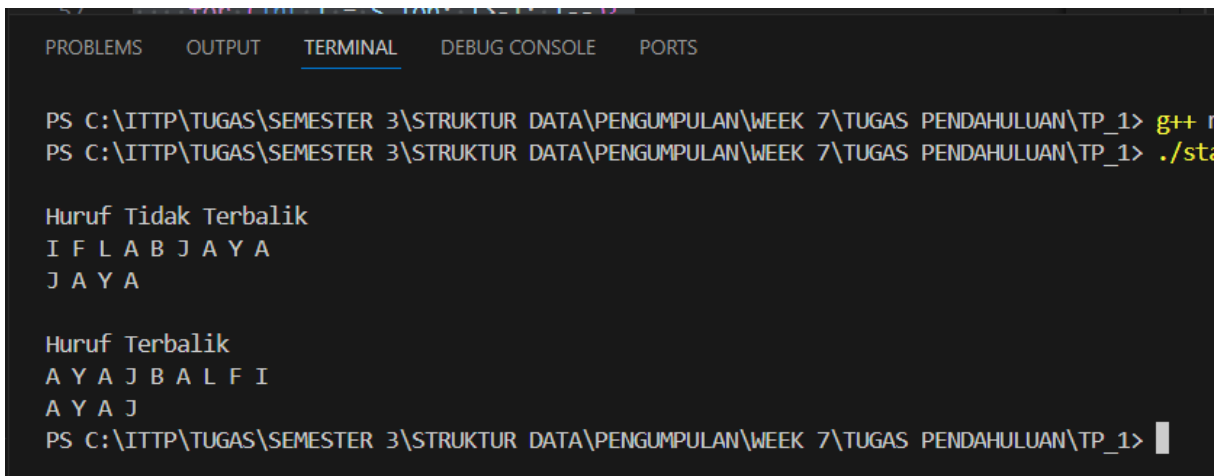
CLASS Stack.cpp

```
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     if(S.Top == 0){
11         return true;
12     }else{
13         return false;
14     }
15 }
16
17 bool isFull(Stack S) {
18     if(S.Top == MAX){
19         return true;
20     }else{
21         return false;
22     }
23 }
24
25 void push(Stack &S, char element) {
26     if (!isFull(S)) {
27         S.Top++;
28         S.info[S.Top] = element;
29     } else {
30         cout << "Stack penuh" << endl;
31     }
32 }
33
34 char pop(Stack &S) {
35     if (!isEmpty(S)) {
36         char element = S.info[S.Top];
37         S.Top--;
38         return element;
39     } else {
40         cout << "Stack kosong" << endl;
41         return '\0';
42     }
43 }
44
45 void printInfo(Stack S) {
46     for (int i = 1; i <= S.Top; i++) {
47         cout << S.info[i]<<" ";
48     }
49     cout << endl;
50 }
```

CLASS main.cpp

```
1 // main.cpp
2 #include "stack.h"
3 #include <iostream>
4 using namespace std;
5
6 int main() {
7     Stack S;
8     createStack(S);
9     cout<<endl;
10    cout<<"Huruf Tidak Terbalik"<<endl;
11    // Menambahkan karakter ke dalam stack
12    char elements[MAX] = {'I', 'F', 'L', 'A', 'B', 'J', 'A', 'Y', 'A'};
13    for (int i = 0; i < 9; i++) {
14        push(S, elements[i]);
15    }
16
17    // Menampilkan isi stack pertama kali
18    printInfo(S);
19
20    // Mengosongkan stack
21    while (!isEmpty(S)) {
22        pop(S);
23    }
24
25    // Menambahkan kembali elemen-elemen tertentu
26    push(S, 'J');
27    push(S, 'A');
28    push(S, 'Y');
29    push(S, 'A');
30
31    // Menampilkan isi stack akhir
32    printInfo(S);
33    while (!isEmpty(S)) {
34        pop(S);
35    }
36    cout<<endl<<"Huruf Terbalik"<<endl;
37    // membuat huruf terbalik
38    char kebalik[MAX] = {'I', 'F', 'L', 'A', 'B', 'J', 'A', 'Y', 'A'};
39    for (int i = 0; i < 9; i++) {
40        push(S, kebalik[i]);
41    }
42
43    for (int i = S.Top; i>-1; i--){
44        cout<<S.info[i]<<" ";
45    }
46
47    while (!isEmpty(S)) {
48        pop(S);
49    }
50
51    push(S, 'J');
52    push(S, 'A');
53    push(S, 'Y');
54    push(S, 'A');
55    cout<<endl;
56
57    for (int i = S.Top; i>-1; i--){
58        cout<<S.info[i]<<" ";
59    }
60    return 0;
61 }
62
```

OUTPUT:



```
PS C:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA\PENGUMPULAN\WEEK 7\TUGAS PENDAHULUAN\TP_1> g++ r
PS C:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA\PENGUMPULAN\WEEK 7\TUGAS PENDAHULUAN\TP_1> ./sta

Huruf Tidak Terbalik
I F L A B J A Y A
J A Y A

Huruf Terbalik
A Y A J B A L F I
A Y A J
PS C:\ITTP\TUGAS\SEMESTER 3\STRUKTUR DATA\PENGUMPULAN\WEEK 7\TUGAS PENDAHULUAN\TP_1> |
```

PENJELASAN

stack.h

Pada kode stack.h di atas mendefinisikan struktur data Stack di C++ dengan kapasitas maksimum 15 elemen bertipe karakter. Terdapat struktur Stack yang memiliki dua anggota: array info untuk menyimpan elemen dan integer Top yang menunjukkan indeks elemen teratas. Kode ini juga mendeklarasikan beberapa fungsi: createStack untuk menginisialisasi stack, isEmpty dan isFull untuk memeriksa status stack, push untuk menambahkan elemen, pop untuk menghapus dan mengembalikan elemen teratas, serta printInfo untuk mencetak semua elemen dalam stack. Fungsi-fungsi ini memberikan kemampuan dasar untuk mengelola stack dan memastikan operasinya dilakukan dengan benar.

stack.cpp

Kode stack.cpp di atas merupakan implementasi fungsi-fungsi untuk mengelola struktur data **Stack** dalam C++. Fungsi createStack menginisialisasi stack dengan mengatur Top menjadi 0, menandakan stack kosong. Fungsi isEmpty memeriksa apakah stack kosong dengan memeriksa nilai Top, sementara isFull mengecek apakah stack telah mencapai kapasitas maksimum. Fungsi push menambahkan elemen ke stack setelah memeriksa apakah stack tidak penuh, sedangkan pop menghapus dan mengembalikan elemen teratas setelah memastikan stack tidak kosong. Terakhir, fungsi printInfo mencetak semua elemen dalam stack. Implementasi ini mencakup penanganan kondisi penuh dan kosong serta mencetak isi stack dengan cara yang terstruktur.

Main.cpp

Kode di atas adalah implementasi program C++ yang menggunakan struktur data **Stack** untuk menampilkan huruf dalam urutan aslinya dan dalam urutan terbalik. Program dimulai dengan membuat objek Stack bernama S, yang kemudian diinisialisasi dengan createStack(). Program pertama menampilkan judul "Huruf Tidak Terbalik", lalu mengisi stack dengan karakter dari array elements, dan menampilkan isinya menggunakan printInfo(). Setelah itu, stack dikosongkan. Kemudian, beberapa karakter baru dimasukkan dan ditampilkan kembali. Selanjutnya, program menampilkan "Huruf Terbalik" dan memasukkan karakter ke dalam stack lagi dari array kebalik, tetapi kali ini mencetaknya dalam urutan terbalik (dari indeks Top hingga 0) untuk menampilkan susunan huruf yang terbalik. Setelah membersihkan stack, beberapa elemen ditambahkan kembali dan dicetak dalam urutan terbalik. Program ini memperlihatkan bagaimana stack dapat digunakan untuk membalikkan urutan elemen melalui penghapusan dan pencetakan dari atas ke bawah.

KODE UNGUIDED

```
1  #include "stack.h"
2  #include <iostream>
3  using namespace std;
4
5  // Fungsi untuk menginisialisasi stack
6  void createStack(Stack &S) {
7      S.top = -1; // Menandakan stack kosong
8  }
9
10 // Fungsi untuk menambahkan elemen ke dalam stack
11 void push(Stack &S, infotype x) {
12     if (S.top < 19) { // Cek apakah stack sudah penuh
13         S.top++;
14         S.array[S.top] = x; // Tambahkan elemen ke stack
15     } else {
16         cout << "Stack sudah penuh" << endl;
17     }
18 }
19
20 // Fungsi untuk mengeluarkan elemen teratas dari stack
21 infotype pop(Stack &S) {
22     if (S.top >= 0) { // Cek apakah stack tidak kosong
23         infotype nilaiArray = S.array[S.top];
24         S.top--;
25         return nilaiArray; // Kembalikan elemen yang dikeluarkan
26     } else {
27         cout << "Stack Kosong" << endl;
28         return -1; // Menandakan tidak ada elemen untuk dikeluarkan
29     }
30 }
31
32 // Fungsi untuk menambahkan elemen ke dalam stack dalam urutan ascending
33 void pushAscending(Stack &S, int x) {
34     // Cek apakah stack sudah penuh
35     if (S.top >= 20 - 1) {
36         cout << "Stack sudah penuh" << endl;
37         return;
38     }
39
40     // Temukan posisi untuk menyisipkan elemen x
41     int position = 0;
42     while (position <= S.top && S.array[position] < x) {
43         position++; // Maju ke elemen berikutnya
44     }
45
46     // Geser elemen yang lebih besar ke kanan
47     for (int i = S.top; i >= position; i--) {
48         S.array[i + 1] = S.array[i]; // Geser elemen ke kanan
49     }
50
51     // Sisipkan elemen x ke posisi yang ditemukan
52     S.array[position] = x;
53     S.top++; // Naikkan indeks top
54 }
```

```

55
56 // Prosedur pengguna memasukkan angka
57 void getInputStream(Stack &S){
58     int banyakBilangan; // menentukan panjang pada array
59     int angka; // menyimpan angka setiap for
60     cout<<"Masukkan banyak bilangan"<<endl;
61     cin>>banyakBilangan;
62
63     if(banyakBilangan >21){ // ketika pengguna memasukkan angka lebih dari panjang top yang telah ditentukan maka akan berhenti
64         cout<<"banyak bilangan melebihi batas"<<endl;
65     }else{
66         for(int i = 0; i<banyakBilangan; i++){
67             cout<<"Masukkan angka "<<i+1<<" = ";
68             cin>>angka; // looping untuk memasukkan angka sesuai kemasukan pengguna
69             S.array[i] = angka;
70             S.top++;
71         }
72     }
73
74 }
75
76 // Fungsi untuk mencetak isi stack
77 void printInfo(Stack &S) {
78     if (S.top == -1) {
79         cout << "Stack Kosong" << endl;
80     } else {
81         for (int i = S.top; i >= 0; i--) {
82             cout << S.array[i] << " "; // Cetak elemen dari atas ke bawah
83         }
84     }
85     cout << endl; // Pindah ke baris baru setelah mencetak
86 }
87
88 // Fungsi untuk mencetak isi stack dari bawah ke atas
89 void balikStack(Stack &S) {
90     if (S.top == -1) {
91         cout << "Stack Kosong" << endl;
92     } else {
93         for (int i = 0; i <= S.top; i++) {
94             cout << S.array[i] << " "; // Cetak elemen dari bawah ke atas
95         }
96     }
97     cout << endl; // Pindah ke baris baru setelah mencetak
98 }
99
100
101

```