

LAPORAN PRAKTIKUM

“Pertemuan ke-6: POST TEST - STACK DENGAN ARRAY”

Diajukan untuk memenuhi salah satu praktikum Mata Struktur Data Informatika yang di
ampu oleh:

Dr., Ardiansyah, S.T., M.Cs.



Disusun Oleh:

Mohammad Farid Hendianto 2200018401

A / Rabu 10.30 – 13.30 Lab. Jaringan

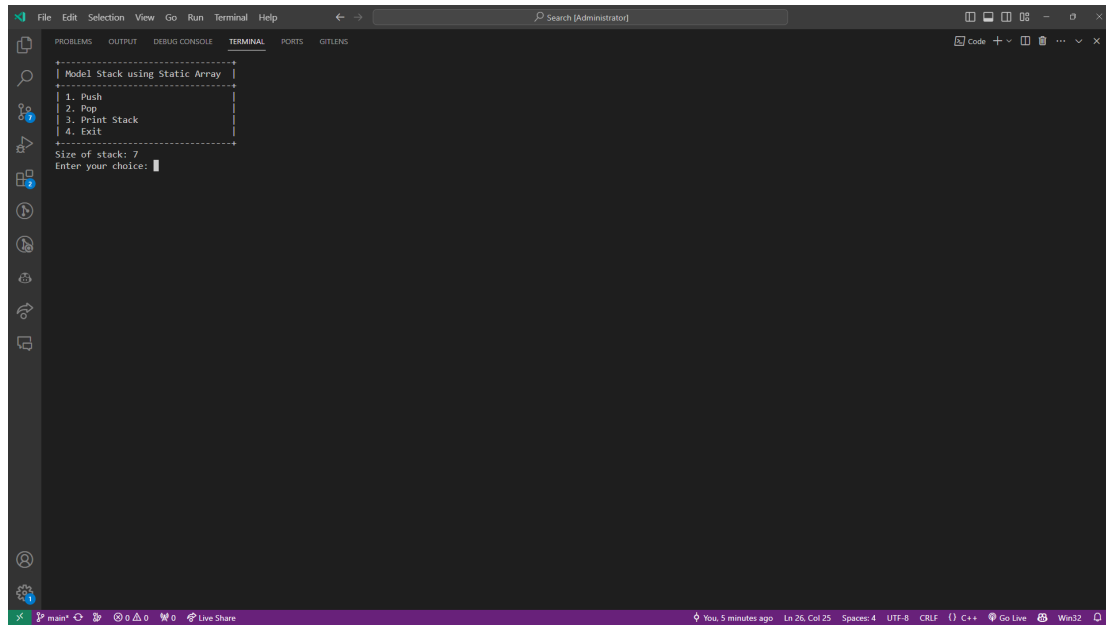
PROGRAM STUDI INFORMATIKA
UNIVERSITAS AHMAD DAHLAN
FAKULTAS TEKNOLOGI INDUSTRI
TAHUN 2023

Buatlah model stack menggunakan array :

a. no ganjil : array statis

Program jalan (Screenshot input output)

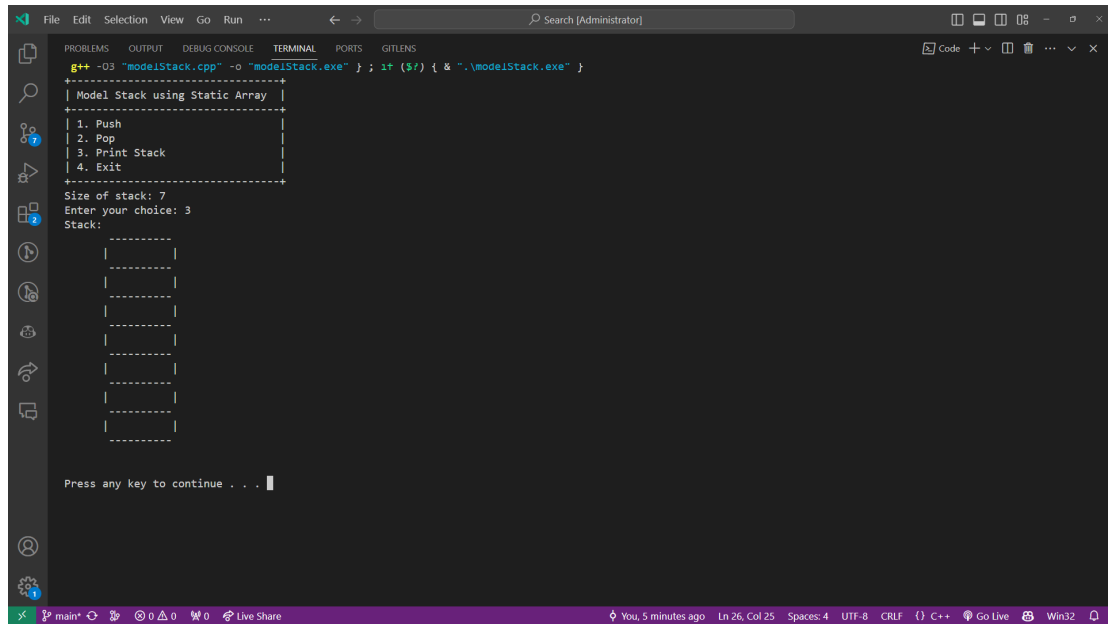
Tampiiilan menu



```

File Edit Selection View Go Run Terminal Help
Model Stack using Static Array
1. Push
2. Pop
3. Print Stack
4. Exit
Size of stack: 7
Enter your choice: 
    
```

Jika Stack masih kosong



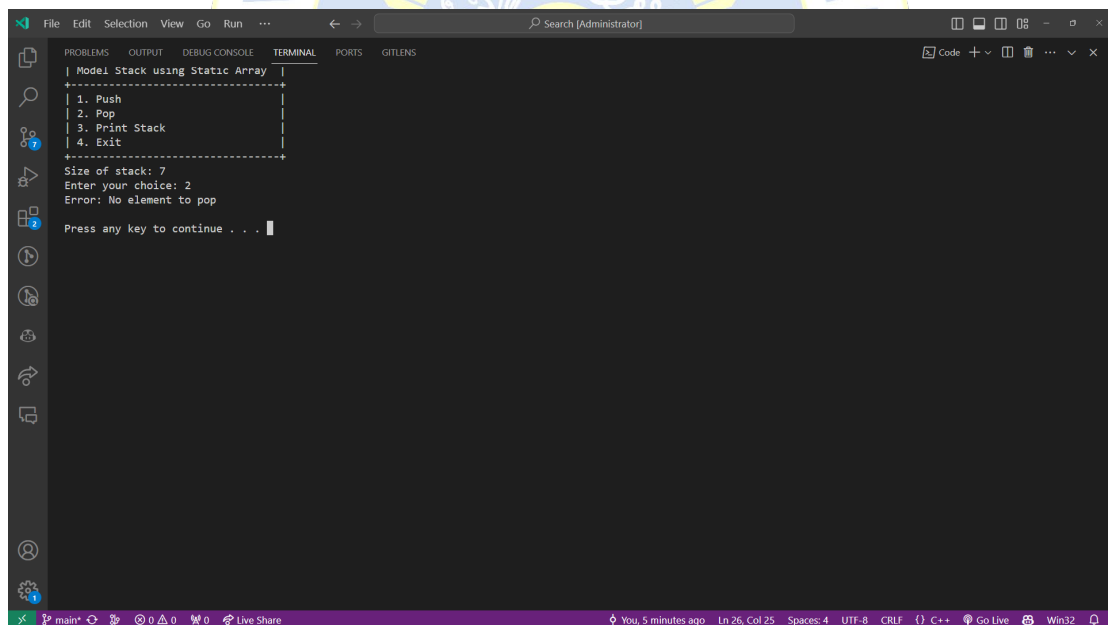
```

g++ -O3 "modelStack.cpp" -o "modelStack.exe" ; if ($?) { & ".\modelStack.exe" }

Model Stack using Static Array
+-----+
| 1. Push |
| 2. Pop  |
| 3. Print Stack |
| 4. Exit |
+-----+
Size of stack: 7
Enter your choice: 3
Stack:
-----
-----
-----
-----
-----
-----
-----

Press any key to continue . . .
  
```

Melakukan pop saat stack kosong



```

Model Stack using Static Array
+-----+
| 1. Push |
| 2. Pop  |
| 3. Print Stack |
| 4. Exit |
+-----+
Size of stack: 7
Enter your choice: 2
Error: No element to pop

Press any key to continue . . .
  
```

Push (5)

```

Model Stack using Static Array
+-----+
| 1. Push |
| 2. Pop  |
| 3. Print Stack |
| 4. Exit |
+-----+
Size of stack: 7
Enter your choice: 1
Enter element to push: 5
Stack:
[ ]
[ ]
[ ]
[ ]
[ ]
[ ]
[ ]

PUSH-> 5

Press any key to continue . . .

```

Push(28)

```

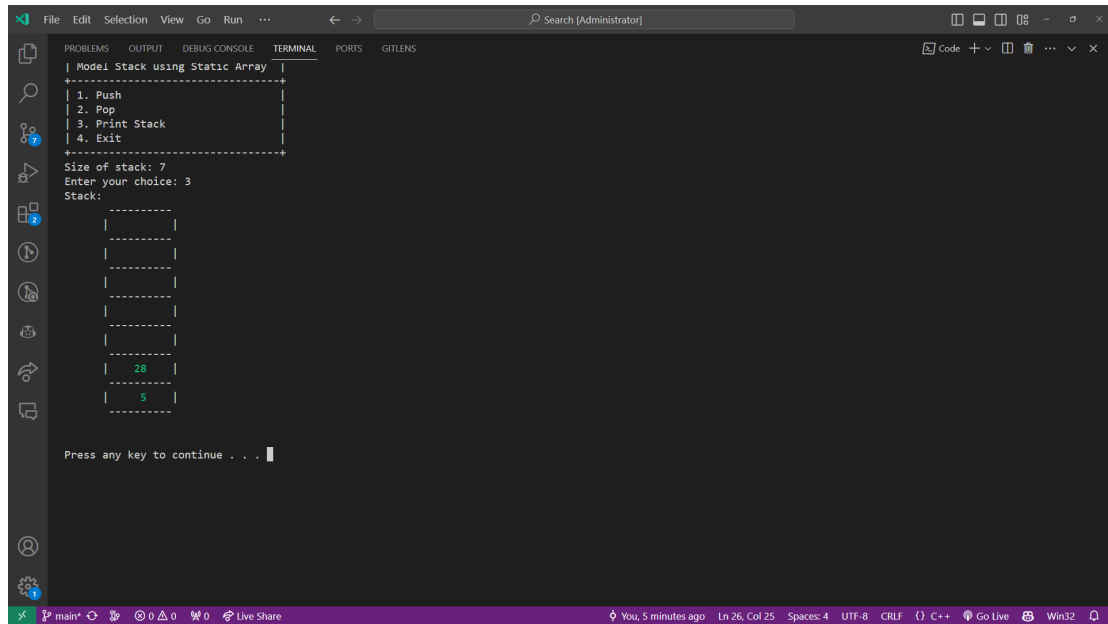
Model Stack using Static Array
+-----+
| 1. Push |
| 2. Pop  |
| 3. Print Stack |
| 4. Exit |
+-----+
Size of stack: 7
Enter your choice: 1
Enter element to push: 28
Stack:
[ ]
[ ]
[ ]
[ ]
[ ]
[ ]
[ 28 ]

PUSH-> 28

Press any key to continue . . .

```

Melihat sementara isi stack semuanya

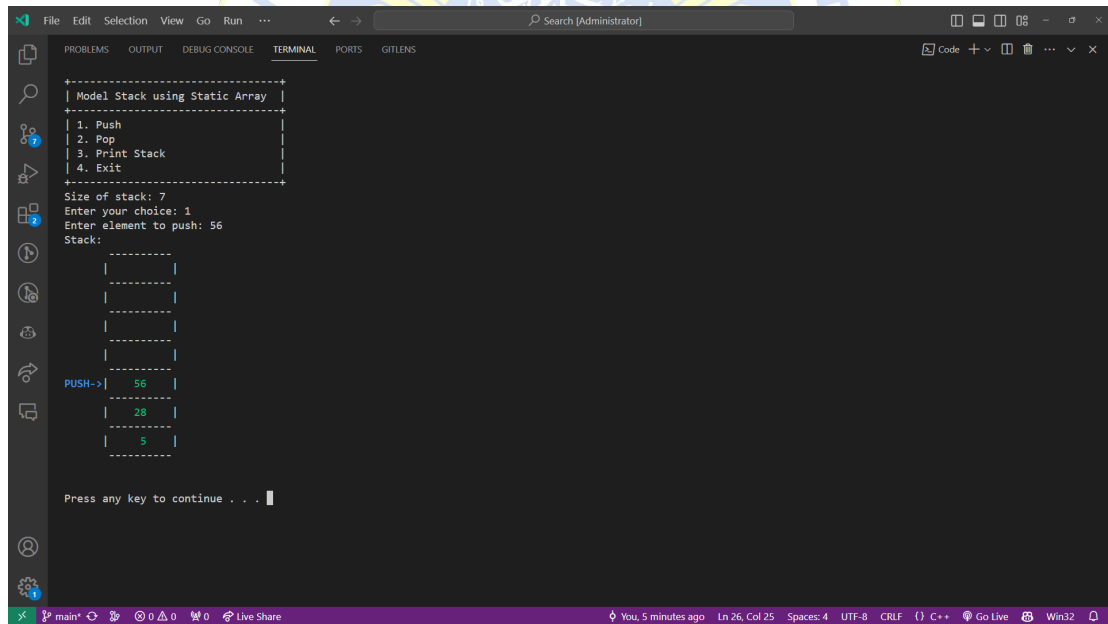


```

Model Stack using Static Array
+-----+
| 1. Push |
| 2. Pop  |
| 3. Print Stack |
| 4. Exit |
+-----+
Size of stack: 7
Enter your choice: 3
Stack:
+-----+
|         |
|         |
|         |
|         |
|         |
| 28      |
| 5       |
|         |
+-----+

Press any key to continue . . .
  
```

Push(56)



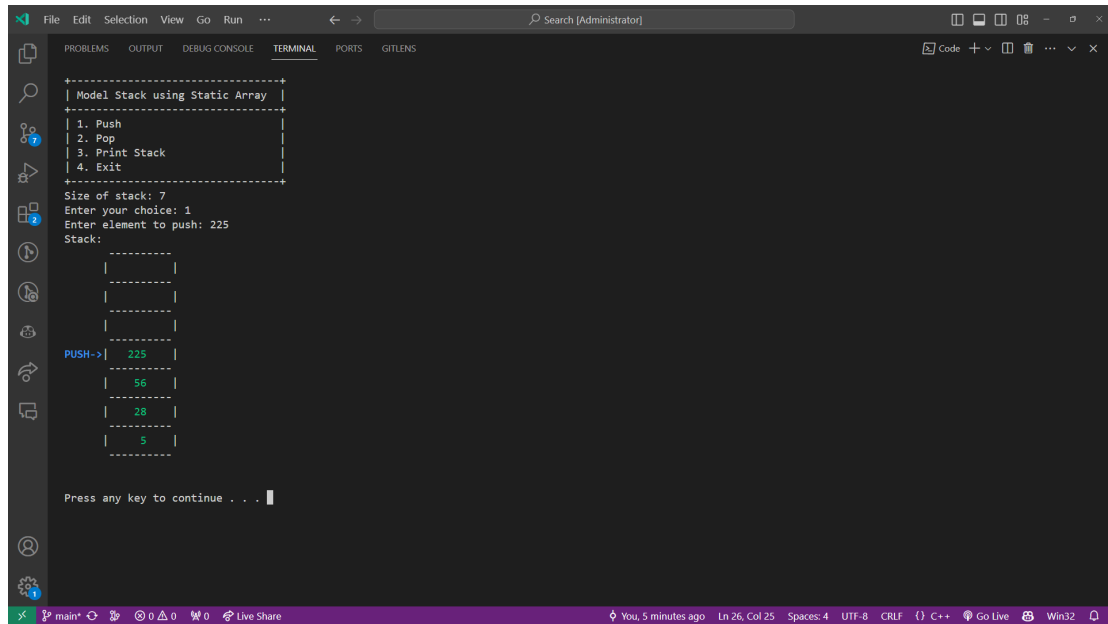
```

Model Stack using Static Array
+-----+
| 1. Push |
| 2. Pop  |
| 3. Print Stack |
| 4. Exit |
+-----+
Size of stack: 7
Enter your choice: 1
Enter element to push: 56
Stack:
+-----+
|         |
|         |
|         |
|         |
|         |
| 56      |
| 28      |
| 5       |
|         |
+-----+

PUSH-> 56

Press any key to continue . . .
  
```

Push(225)



```

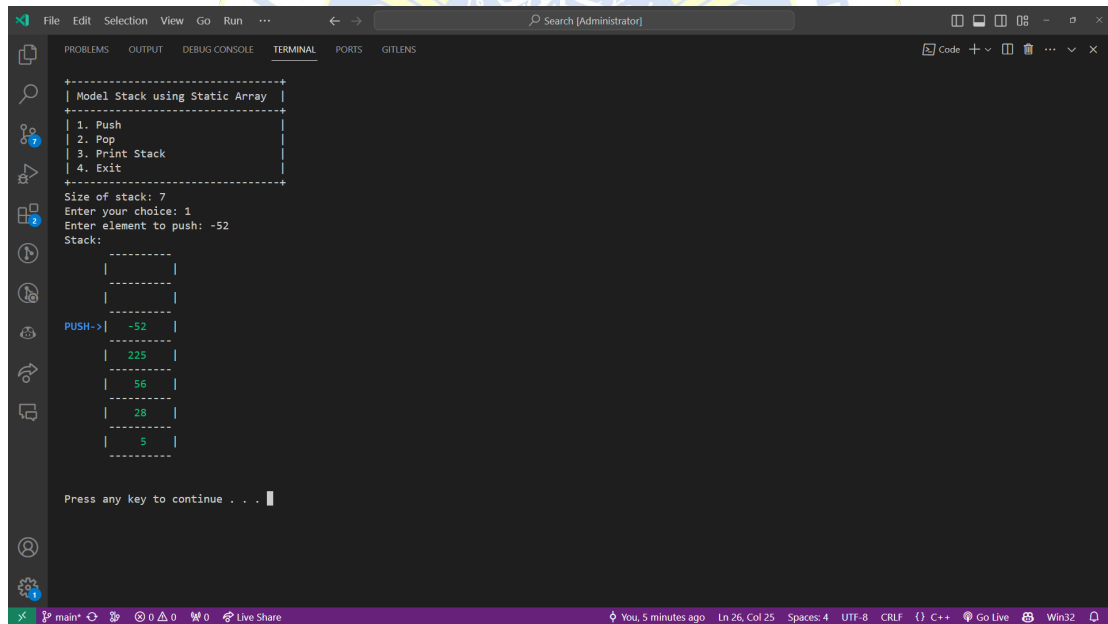
+-----+
| Model Stack using Static Array |
+-----+
| 1. Push
| 2. Pop
| 3. Print Stack
| 4. Exit
+-----+

Size of stack: 7
Enter your choice: 1
Enter element to push: 225
Stack:

+-----+
|         |
|         |
|         |
|         |
| 225     |
| 56      |
| 28      |
| 5       |
+-----+

Press any key to continue . . .
  
```

Push(-52)



```

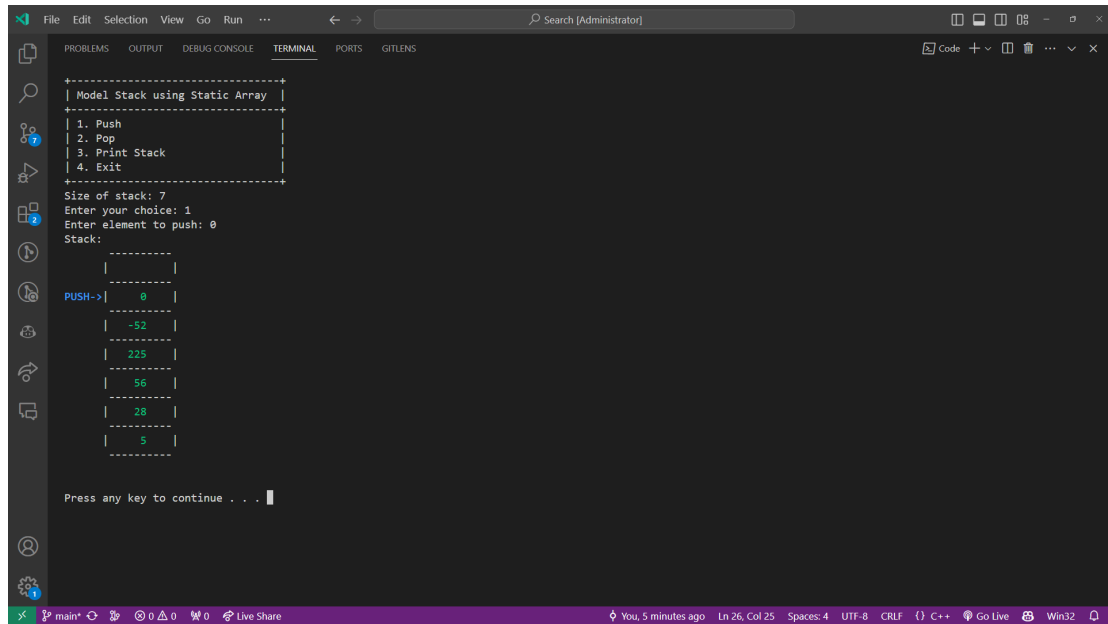
+-----+
| Model Stack using Static Array |
+-----+
| 1. Push
| 2. Pop
| 3. Print Stack
| 4. Exit
+-----+

Size of stack: 7
Enter your choice: 1
Enter element to push: -52
Stack:

+-----+
|         |
|         |
|         |
|         |
| -52     |
| 225     |
| 56      |
| 28      |
| 5       |
+-----+

Press any key to continue . . .
  
```

Push(0)



```

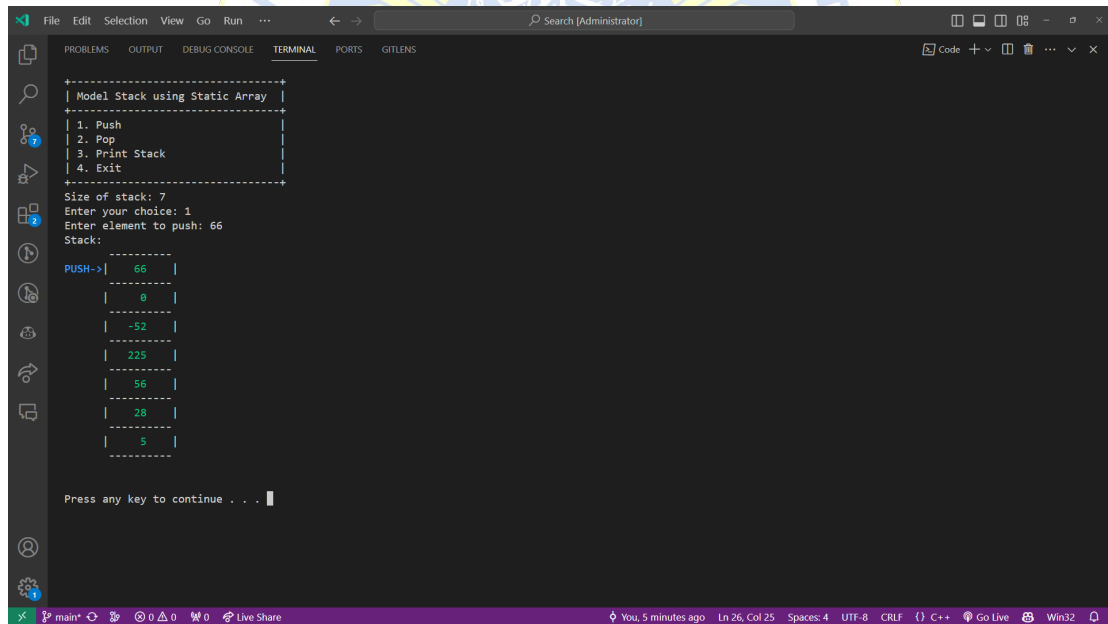
+-----+
| Model Stack using Static Array |
+-----+
| 1. Push
| 2. Pop
| 3. Print Stack
| 4. Exit
+-----+

Size of stack: 7
Enter your choice: 1
Enter element to push: 0
Stack:

-----
| 0 |
| -52 |
| 225 |
| 96 |
| 28 |
| 5 |
-----

Press any key to continue . . .
  
```

Push(66)



```

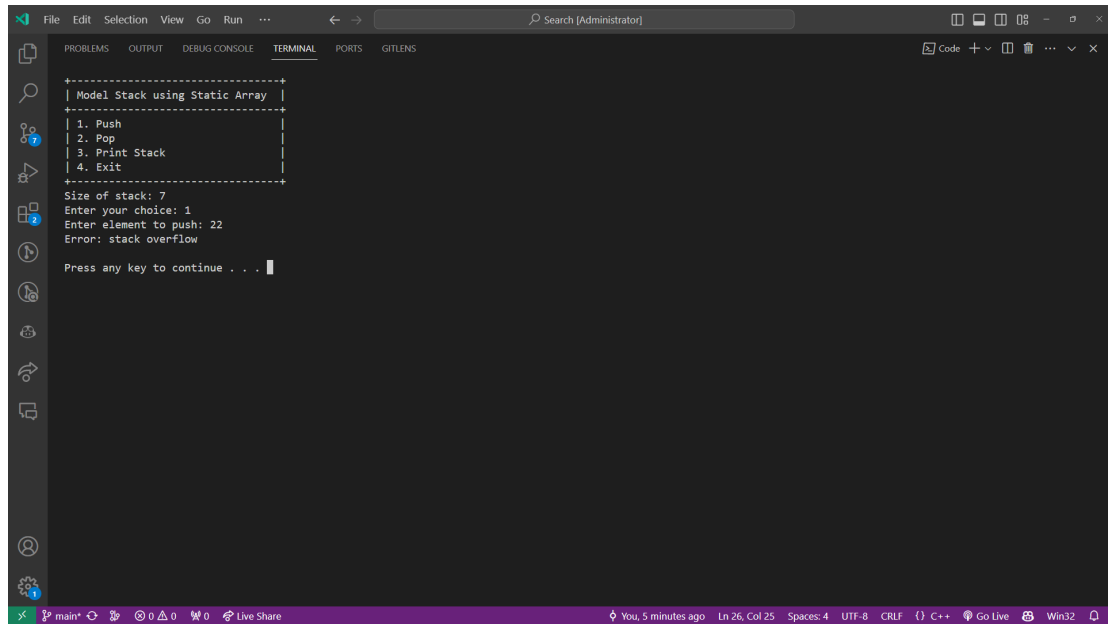
+-----+
| Model Stack using Static Array |
+-----+
| 1. Push
| 2. Pop
| 3. Print Stack
| 4. Exit
+-----+

Size of stack: 7
Enter your choice: 1
Enter element to push: 66
Stack:

-----
| 66 |
| 0 |
| -52 |
| 225 |
| 96 |
| 28 |
| 5 |
-----

Press any key to continue . . .
  
```

Push(22) tetapi karena stack sudah penuh maka akan dikirim pesan

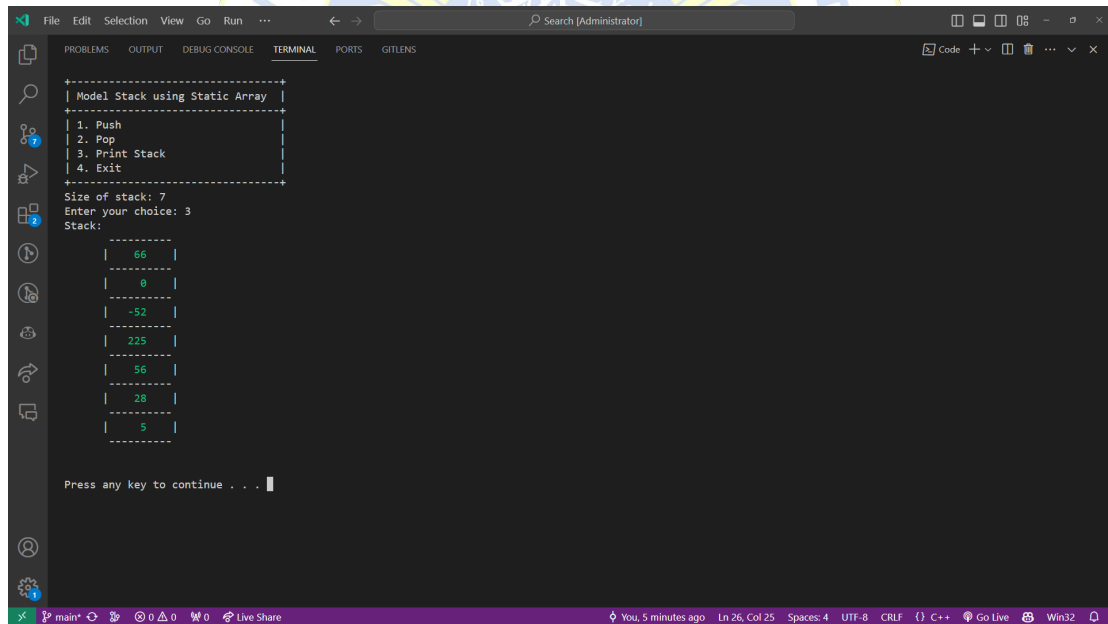


```

+-----+
| Model Stack using Static Array |
+-----+
| 1. Push
| 2. Pop
| 3. Print Stack
| 4. Exit
+-----+
Size of stack: 7
Enter your choice: 1
Enter element to push: 22
Error: stack overflow

Press any key to continue . . .
  
```

Isi stack sekarang

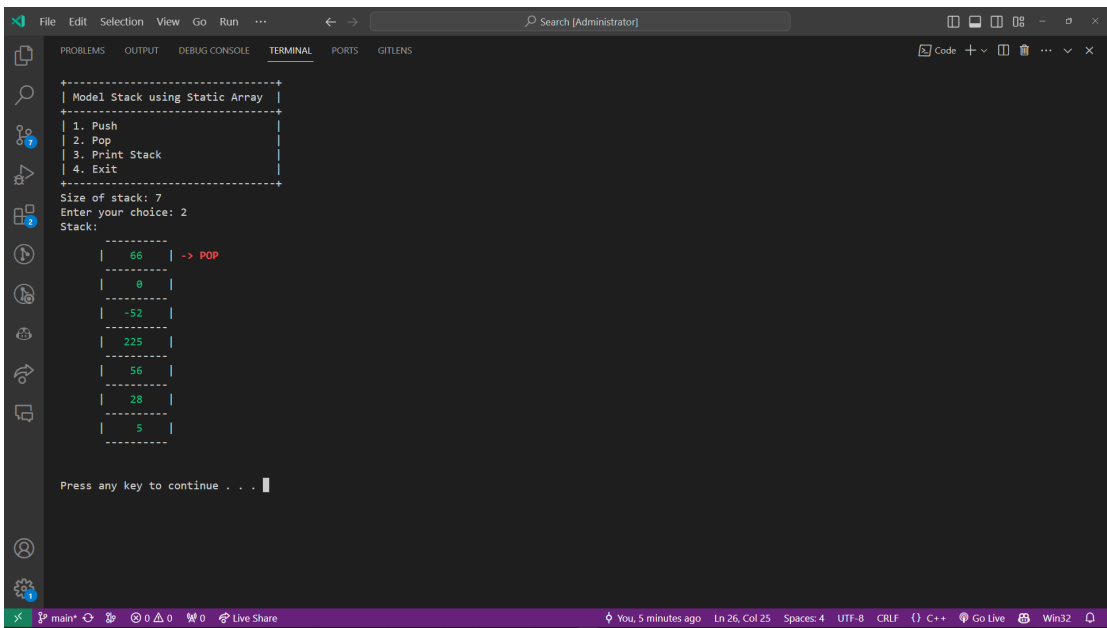


```

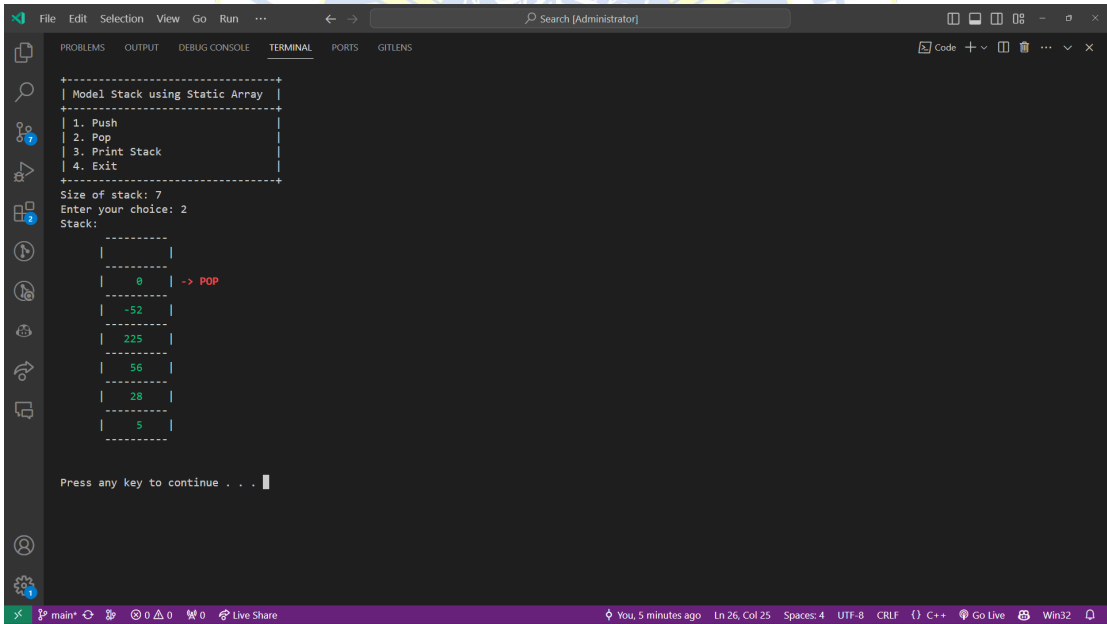
+-----+
| Model Stack using Static Array |
+-----+
| 1. Push
| 2. Pop
| 3. Print Stack
| 4. Exit
+-----+
Size of stack: 7
Enter your choice: 3
Stack:
-----
| 66 |
| 0 |
| -52 |
| 225 |
| 96 |
| 28 |
| 5 |
-----

Press any key to continue . . .
  
```

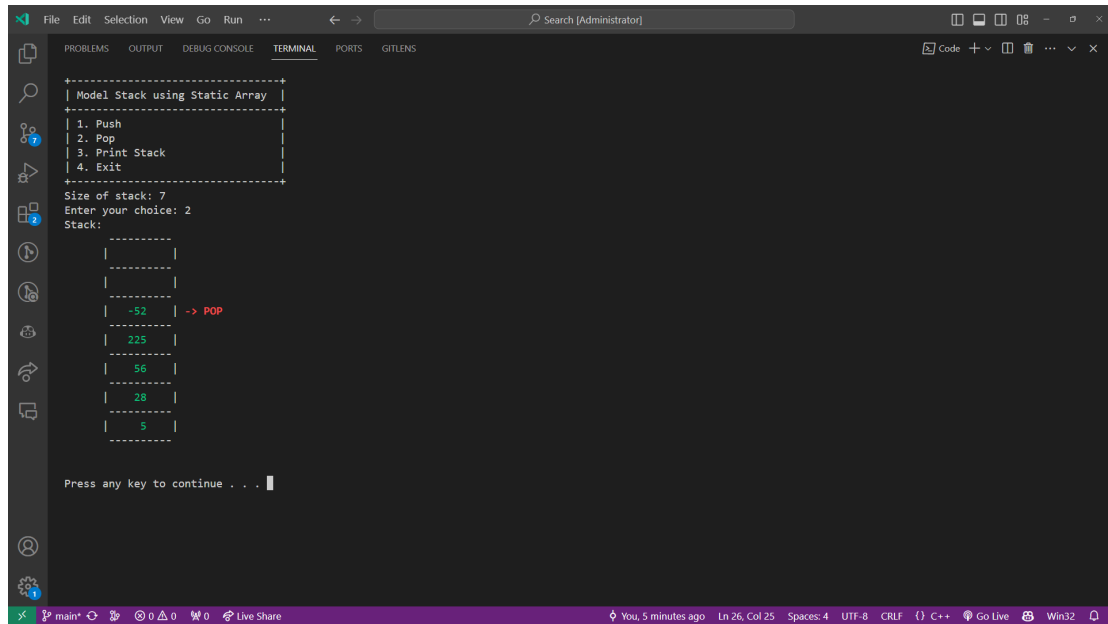
Pop()



Pop()



Pop()



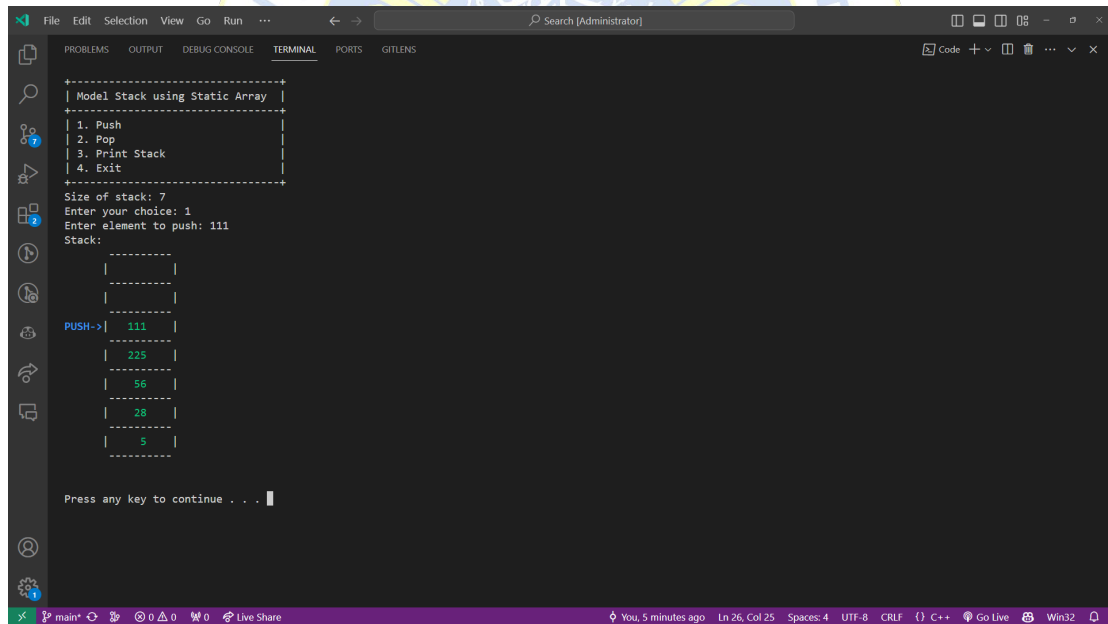
```

+-----+
| Model Stack using Static Array |
+-----+
| 1. Push |
| 2. Pop  |
| 3. Print Stack |
| 4. Exit |
+-----+

Size of stack: 7
Enter your choice: 2
Stack:
+-----+
|         |
|         |
|        -52 | -> POP
|        225 |
|         96 |
|         28 |
|          5 |
+-----+

Press any key to continue . . .
  
```

Push(111)



```

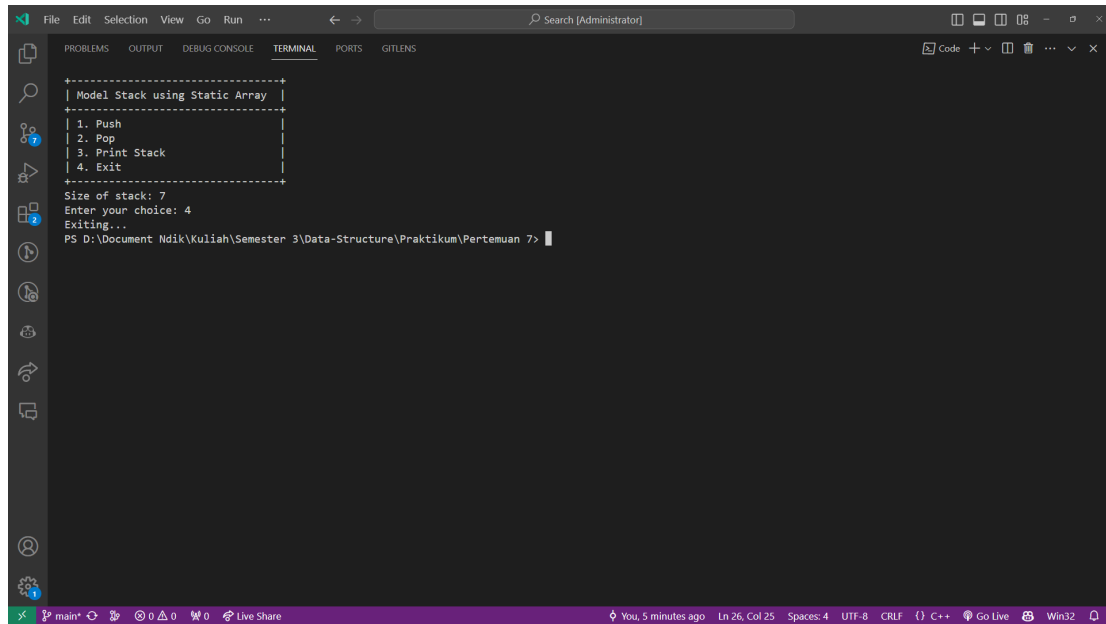
+-----+
| Model Stack using Static Array |
+-----+
| 1. Push |
| 2. Pop  |
| 3. Print Stack |
| 4. Exit |
+-----+

Size of stack: 7
Enter your choice: 1
Enter element to push: 111
Stack:
+-----+
|         |
|         |
|         |
|        225 |
|         96 |
|         28 |
|          5 |
+-----+

PUSH->| 111 |
+-----+

Press any key to continue . . .
  
```

Keluar program



```

File Edit Selection View Go Run ... Search [Administrator]
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS
| Model Stack using Static Array |
| 1. Push |
| 2. Pop |
| 3. Print Stack |
| 4. Exit |
Size of stack: 7
Enter your choice: 4
Exiting...
PS D:\Document Ndik\Kuliah\Semester 3\Data-Structure\Praktikum\Pertemuan 7>

```

Operasi menambah data (push)



Fungsi pop()

Fungsi push dalam kode ini digunakan untuk menambahkan elemen ke tumpukan (stack) dalam bahasa pemrograman C++. Berikut adalah penjelasan langkah demi langkah:

Fungsi push menerima satu parameter, yaitu x, yang merupakan nilai yang akan ditambahkan ke tumpukan.

Pertama, fungsi ini memeriksa apakah tumpukan sudah penuh. Ini dilakukan dengan membandingkan variabel top dengan MAX_SIZE - 1. top adalah indeks elemen teratas dalam tumpukan, dan MAX_SIZE adalah kapasitas maksimum tumpukan. Jika top sama dengan MAX_SIZE - 1, ini berarti tumpukan sudah penuh.

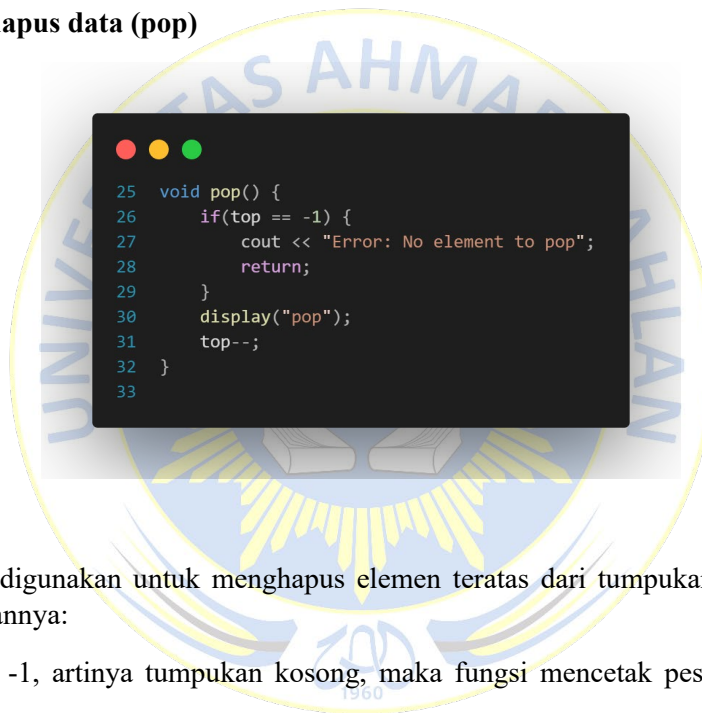
Jika tumpukan sudah penuh, fungsi ini mencetak pesan kesalahan "Error: stack overflow" dan kemudian mengakhiri eksekusi dengan return.

Jika tumpukan belum penuh, fungsi ini menambahkan nilai x ke tumpukan. Ini dilakukan dengan pertama-tama menambahkan 1 ke top (dengan ++top), dan kemudian menetapkan A[top] menjadi x. A adalah array yang digunakan untuk menyimpan elemen-elemen tumpukan.

Akhirnya, fungsi ini memanggil fungsi display dengan argumen "push" sebagai penampilan step by step push.

Untuk contoh operasi dapat melihat di screenshot.

Operasi menghapus data (pop)



Fungsi pop ini digunakan untuk menghapus elemen teratas dari tumpukan (stack). Berikut adalah penjelasannya:

Jika top adalah -1, artinya tumpukan kosong, maka fungsi mencetak pesan kesalahan dan berhenti.

Jika tidak, fungsi memanggil display("pop") (kemungkinan untuk mencetak status tumpukan) dan mengurangi top sebanyak 1, yang efektif menghapus elemen teratas dari tumpukan.

Untuk contoh operasi dapat melihat di screenshot.

Class

Class Stack

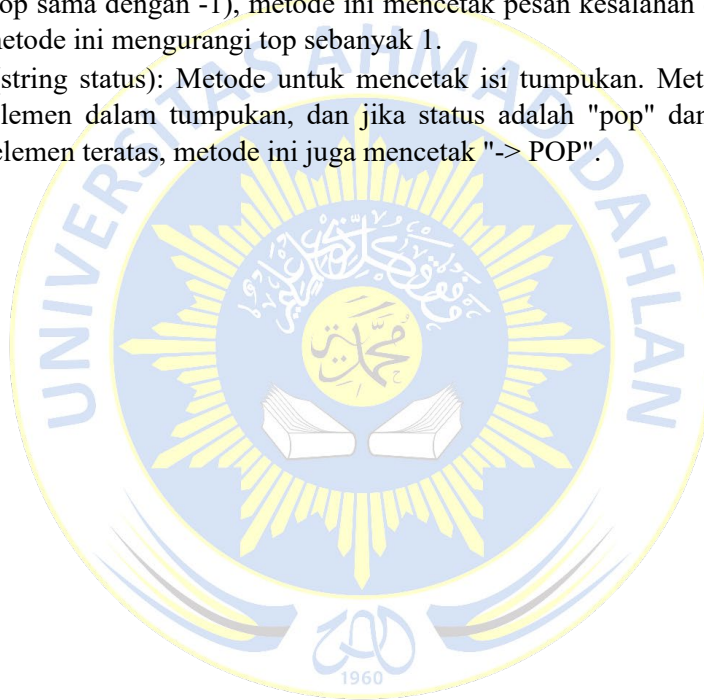
```

5  class Stack {
6  private:
7      int A[MAX_SIZE];
8      int top;
9
10 public:
11     Stack() {
12         top = -1;
13     }
14
15     void push(int x) {
16         if(top == MAX_SIZE - 1) {
17             cout << "Error: stack overflow";
18             return;
19         }
20         A[++top] = x;
21         display("push");
22     }
23
24     void pop() {
25         if(top == -1) {
26             cout << "Error: No element to pop";
27             return;
28         }
29         display("pop");
30         top--;
31     }
32
33     int Top() {
34         return A[top];
35     }
36
37     bool isEmpty() {
38         if(top == -1) {
39             return true;
40         }
41         return false;
42     }
43
44     void display(){
45         display("");
46     }
47     void display(string status) {
48         string border = "-----";
49         string blank = "      |      ";
50
51         cout << "Stack: " << endl;
52         for (int i = MAX_SIZE - 1; i >= 0; i--) {
53             cout << "      " << border << endl;
54             if (i > top) {
55                 cout << blank << endl;
56                 continue;
57             }
58             string label = (status == "push" && i == top) ? "\033[1;34mPUSH->\033[0m" : "      ";
59             cout << label << " | \033[32m" << setw(5) << A[i] << "\033[0m      |";
60             if (status == "pop" && i == top)
61                 cout << "\033[1;31m -> POP\033[0m";
62             cout << endl;
63         }
64         cout << "      " << border << endl;
65     }
66 };

```

Class Stack dalam kode ini adalah implementasi dari struktur data tumpukan (stack) menggunakan array statis. Berikut adalah penjelasannya:

- A: Array yang digunakan untuk menyimpan elemen-elemen tumpukan. Tipe data dan ukuran array tidak ditunjukkan dalam potongan kode ini.
- top: Variabel yang digunakan untuk melacak indeks elemen teratas dalam tumpukan. Biasanya, top diatur menjadi -1 ketika tumpukan kosong.
- push(int x): Metode untuk menambahkan elemen ke tumpukan. Jika tumpukan sudah penuh (yaitu, top sama dengan MAX_SIZE - 1), metode ini mencetak pesan kesalahan dan berhenti. Jika tidak, metode ini menambahkan x ke tumpukan dan memperbarui top.
- pop(): Metode untuk menghapus elemen teratas dari tumpukan. Jika tumpukan kosong (yaitu, top sama dengan -1), metode ini mencetak pesan kesalahan dan berhenti. Jika tidak, metode ini mengurangi top sebanyak 1.
- display(string status): Metode untuk mencetak isi tumpukan. Metode ini mencetak setiap elemen dalam tumpukan, dan jika status adalah "pop" dan elemen tersebut adalah elemen teratas, metode ini juga mencetak "-> POP".



Class Menu

```

68 class Menu {
69 private:
70     Stack S;
71 public:
72     void displayMenu() {
73         int choice;
74         while (true){
75             cout << "+-----+"<<endl;
76             cout << "| Model Stack using Static Array |"<<endl;
77             cout << "+-----+"<<endl;
78             cout << "| 1. Push                               |" << endl;
79             cout << "| 2. Pop                               |" << endl;
80             cout << "| 3. Print Stack                       |" << endl;
81             cout << "| 4. Exit                               |" << endl;
82             cout << "+-----+"<<endl;
83             cout << "Size of stack: " << MAX_SIZE << endl;
84             cout << "Enter your choice: ";
85             cin >> choice;
86             switch(choice) {
87                 case 1:
88                     int x;
89                     cout << "Enter element to push: ";
90                     cin >> x;
91                     S.push(x);
92                     break;
93                 case 2:
94                     S.pop();
95                     break;
96                 case 3:
97                     S.display();
98                     break;
99                 case 4:
100                     cout << "Exiting..." << endl;
101                     exit(0);
102                     break;
103                 default:
104                     cout << "Invalid choice" << endl;
105             }
106             cout<<endl<<endl;
107             system("Pause");system("cls");
108         }
109     }
110 };

```

Class Menu adalah sebuah class dalam C++ yang saat ini masih kosong. Class ini memiliki satu private member, yaitu objek S dari class Stack. Fungsi dan metode dari class Menu belum didefinisikan dalam potongan kode yang diberikan. Biasanya, class seperti ini akan memiliki metode untuk berinteraksi dengan objek Stack S, seperti menambahkan item ke stack, menghapus item, atau menampilkan isi stack.

Source Code

```

1  #include <iostream>
2  #include <iomanip>
3  #define MAX_SIZE 7
4  using namespace std;
5  class Stack {
6  private:
7      int A[MAX_SIZE];
8      int top;
9
10 public:
11     Stack() {
12         top = -1;
13     }
14
15     void push(int x) {
16         if(top == MAX_SIZE - 1) {
17             cout << "Error: stack overflow";
18             return;
19         }
20         A[++top] = x;
21         display("push");
22     }
23
24     void pop() {
25         if(top == -1) {
26             cout << "Error: No element to pop";
27             return;
28         }
29         display("pop");
30         top--;
31     }
32
33     int Top() {
34         return A[top];
35     }
36
37     bool isEmpty() {
38         if(top == -1) {
39             return true;
40         }
41         return false;
42     }
43
44     void display(){
45         display("");
46     }
47     void display(string status) {
48         string border = "-----";
49         string blank = "          |          ";
50
51         cout << "Stack: " << endl;
52         for (int i = MAX_SIZE - 1; i >= 0; i--) {
53             cout << "          " << border << endl;
54             if (i > top) {
55                 cout << blank << endl;
56                 continue;
57             }
58             string label = (status == "push" && i == top) ? "\033[1;34mPUSH->\033[0m" : "      ";
59             cout << label << " | \033[32m" << setw(5) << A[i] << "\033[0m   |";
60             if (status == "pop" && i == top)
61                 cout << "\033[1;31m -> POP\033[0m";
62             cout << endl;
63         }
64         cout << "          " << border << endl;
65     }
66 };
67

```



```

68 class Menu {
69 private:
70     Stack S;
71 public:
72     void displayMenu() {
73         int choice;
74         while (true){
75             cout << "+-----+"<<endl;
76             cout << "| Model Stack using Static Array |"<<endl;
77             cout << "+-----+"<<endl;
78             cout << "| 1. Push |" << endl;
79             cout << "| 2. Pop |" << endl;
80             cout << "| 3. Print Stack |" << endl;
81             cout << "| 4. Exit |" << endl;
82             cout << "+-----+"<<endl;
83             cout << "Size of stack: " << MAX_SIZE << endl;
84             cout << "Enter your choice: ";
85             cin >> choice;
86             switch(choice) {
87                 case 1:
88                     int x;
89                     cout << "Enter element to push: ";
90                     cin >> x;
91                     S.push(x);
92                     break;
93                 case 2:
94                     S.pop();
95                     break;
96                 case 3:
97                     S.display();
98                     break;
99                 case 4:
100                     cout << "Exiting..." << endl;
101                     exit(0);
102                     break;
103                 default:
104                     cout << "Invalid choice" << endl;
105             }
106             cout<<endl<<endl;
107             system("Pause");system("cls");
108         }
109     }
110 };
111
112 int main() {
113     Menu menu;
114     menu.displayMenu();
115     return 0;
116 }

```

Untuk melihat source kodingan, dapat melihat link github berikut:

[IRedDragonICY/Data-Structure \(github.com\)](https://github.com/IRedDragonICY/Data-Structure)

