

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL 7
STACK**



Disusun Oleh :
NAMA : Muhammad Azzam Satria
NIM : 103112400112

Dosen
FAHRUDIN MUKTI WIBOWO S. Kom., M. Eng

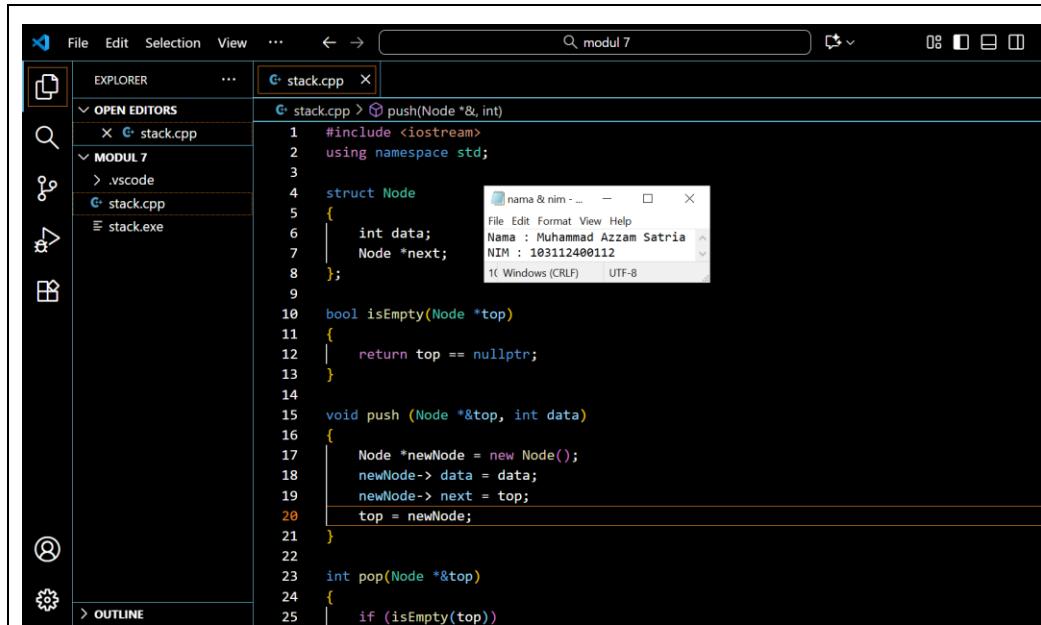
**PROGRAM STUDI STRUKTUR DATA
FAKULTAS INFORMATIKA
TELKOM UNIVERSITY PURWOKERTO
2025**

A. Dasar Teori

Pada masa kini, struktur data merupakan suatu hal yang sangat penting pada pengembangan perangkat lunak. Salah satu struktur data yang relevan dengan zaman sekarang adalah stack. Stack pada bahasa indonesia dapat diartikan sebagai tumpukan. Stack merupakan struktur data linier yang memiliki prinsip last in first out (LIFO), artinya elemen yang terakhir disisipkan akan menjadi elemen pertama yang keluar. Cara kerja seperti ini banyak digunakan di berbagai aplikasi, misalnya evaluasi rumus, manajemen memori dan pemrosesan fungsi secara rekursif.

Selain itu, dalam implementasi stack terdapat 2 operasi kunci yaitu push dan pop. Operasi push memiliki fungsi untuk menambah data ke dalam stack, data baru yang ditambahkan akan menjadi elemen teratas pada stack, sementara itu, operasi pop berfungsi untuk menghapus elemen yang berada di posisi atas, sehingga elemen dibawahnya akan menjadi elemen baru di bagian atas pada stack. Implementasi stack dapat dilakukan dengan cara menggunakan array dan linked list. Dengan implementasi tersebut, stack dapat memberikan fleksibilitas dalam alur kerja program, terutama ketika terdapat proses pemanggilan fungsi berantai dan penyimpanan nilai sementara ketika eksekusi program berlangsung.

B. Guided



```
#include <iostream>
using namespace std;

struct Node
{
    int data;
    Node *next;
};

bool isEmpty(Node *top)
{
    return top == nullptr;
}

void push (Node *&top, int data)
{
    Node *newNode = new Node();
    newNode-> data = data;
    newNode-> next = top;
    top = newNode;
}

int pop(Node *&top)
{
    if (isEmpty(top))
        return -1;
    Node *temp = top;
    top = top-> next;
    delete temp;
    return top-> data;
}
```

File Edit Selection View ... ← → 🔍 modul 7

EXPLORER OPEN EDITORS stack.cpp MODUL 7 .vscode stack.cpp stack.exe

```
26 {  
27 |     cout << "Stack Kosong, tidak bisa pop!" << endl;  
28 |     return 0;  
29 | }  
30 int poppedData = top-> data;  
31 Node *temp = top;  
32 top = top-> next;  
33  
34 delete temp;  
35 return poppedData;  
36 }  
37  
38 void show(Node *top)  
39 {  
40 |     if (isEmpty(top))  
41 |     {  
42 |         cout << "Stack Kosong." << endl;  
43 |         return;  
44 |     }  
45 cout << "TOP ->";  
46 Node *temp = top;  
47  
48 while (temp != nullptr)  
49 {  
50 |     cout << temp-> data << " ->";  
51 |     temp = temp-> next;  
52 }  
53 cout << "NULL" << endl;  
54 }  
55  
56 int main()  
57 {  
58     Node *stack = nullptr;  
59  
60     push (stack, 10);  
61     push (stack, 20);  
62     push (stack, 30);  
63  
64     cout << "Menampilkan isi stack: " << endl;  
65     show(stack);  
66  
67     cout << "Pop: " << pop(stack) << endl;  
68  
69     cout << "Menampilkan sisa stack: " << endl;  
70     show(stack);  
71  
72     return 0;  
73 }
```

nama & nim - File Edit Format View Help
Nama : Muhammad Azzam Satria
NIM : 103112400112
1f Windows (CRLF) UTF-8

File Edit Selection View ... ← → 🔍 modul 7

EXPLORER OPEN EDITORS stack.cpp MODUL 7 .vscode stack.cpp stack.exe

```
52 }  
53 cout << "NULL" << endl;  
54 }  
55  
56 int main()  
57 {  
58     Node *stack = nullptr;  
59  
60     push (stack, 10);  
61     push (stack, 20);  
62     push (stack, 30);  
63  
64     cout << "Menampilkan isi stack: " << endl;  
65     show(stack);  
66  
67     cout << "Pop: " << pop(stack) << endl;  
68  
69     cout << "Menampilkan sisa stack: " << endl;  
70     show(stack);  
71  
72     return 0;  
73 }
```

nama & nim - File Edit Format View Help
Nama : Muhammad Azzam Satria
NIM : 103112400112
1f Windows (CRLF) UTF-8

zScreenshots Output

```

PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\modul 7> & 'c:\Users\USER\.vscode\extensions\ms-vscode.cpptools-1.28.3-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-jh1o4wxm.td0' '--stdout=Microsoft-MIEngine-Out-1zauepu.g31' '--stderr=Microsoft-MIEngine-Error-fin1bsuu.wy0' '--pid=Microsoft-MIEngine-Pid-myus2kie.0b1' '--dbgExe=C:\Users\USER\mingw32\bin\gdb.exe' '--interpreter=mi'
Menampilkan isi stack:
TOP ->30 ->20 ->10 ->NULL
Pop: 30
Menampilkan sisa stack:
TOP ->20 ->10 ->NULL
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\modul 7>

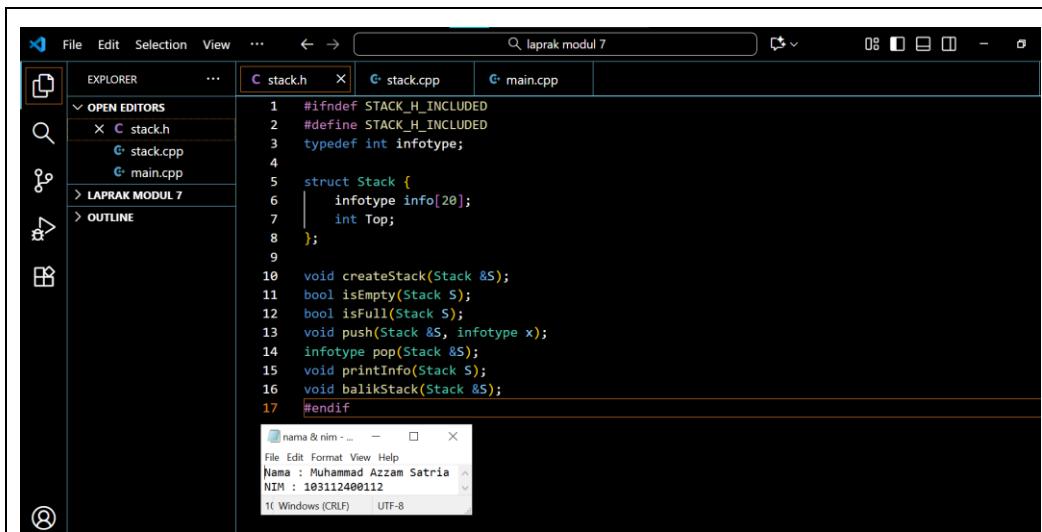
```

Deskripsi

Program ini menggunakan struktur data node yang berisi dua elemen yaitu data bertipe integer dan pointer (next) untuk menghubungkan ke node berikutnya. Program mendeklarasikan beberapa fungsi yaitu isEmpty, push, pop dan show untuk mengelola stack. Fungsi isEmpty yaitu untuk memeriksa apakah stack kosong dengan mengecek pointer top bernilai nullptr. Fungsi push untuk membuat node baru dengan nilai data yang diterima dari parameter, lalu menghubungkan ke node teratas sebelumnya, sehingga node baru menjadi elemen paling atas. Fungsi pop yaitu untuk menghapus elemen paling atas, lalu menyimpan nilainya ke variabel sementara, kemudian menghapus node dari memori menggunakan delete.

Fungsi show yaitu untuk menampilkan seluruh isi stack dengan menelusuri setiap node dari atas hingga akhir dengan mencetak nilai datanya. Pada fungsi main, pointer stack diinisialisasi dengan nilai nullptr sebagai tanda kalau stack masih kosong. Kemudian program melakukan tiga kali operasi push untuk menambahkan nilai 10, 20 dan 30 ke dalam stack. Setelah itu, program akan mencetak output dengan memanggil fungsi show yang menampilkan urutan data dari atas hingga bawah. Lalu fungsi pop akan menghapus elemen teratas dan hasil penghapusan langsung ditampilkan. Selanjutnya, fungsi show akan menampilkan sisa isi stack setelah elemen teratas dihapus.

C. Unguided 1



File Edit Selection View ... ← → Q: laprak modul 7

EXPLORER ... C stack.h X C stack.cpp C main.cpp

OPEN EDITORS

X C stack.h C stack.cpp C main.cpp

> LAPRAK MODUL 7

> OUTLINE

stack.h

```
1 #ifndef STACK_H_INCLUDED
2 #define STACK_H_INCLUDED
3 typedef int infotype;
4
5 struct Stack {
6     infotype info[20];
7     int Top;
8 };
9
10 void createStack(Stack &S);
11 bool isEmpty(Stack S);
12 bool isFull(Stack S);
13 void push(Stack &S, infotype x);
14 infotype pop(Stack &S);
15 void printInfo(Stack S);
16 void balikStack(Stack &S);
17 #endif
```

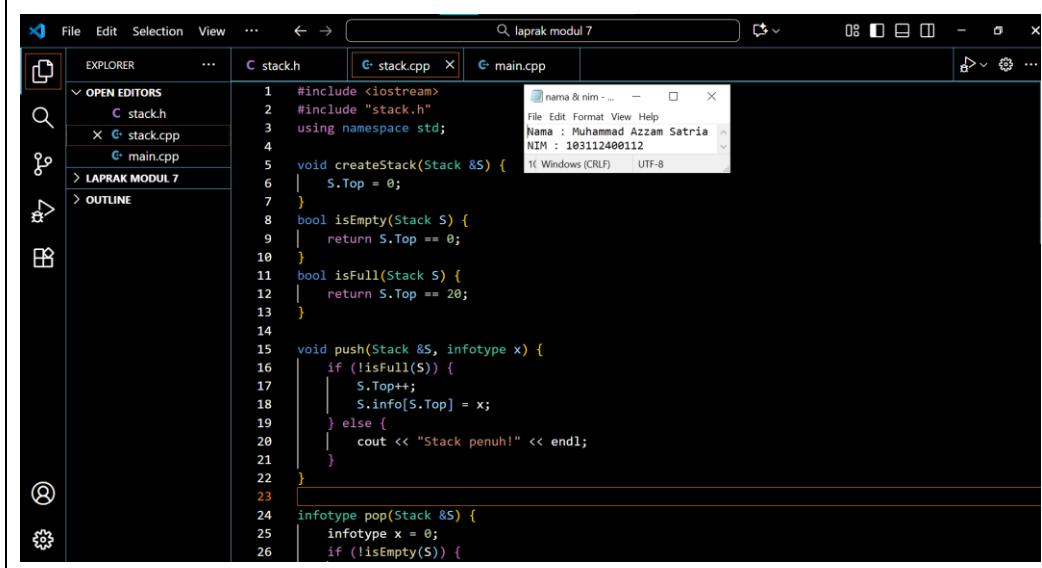
nama & nim - ...

File Edit Format View Help

Nama : Muhammad Azzam Satria

NIM : 103112400112

1f Windows (CRLF) UTF-8



File Edit Selection View ... ← → Q: laprak modul 7

EXPLORER ... C stack.h X C stack.cpp C main.cpp

OPEN EDITORS

C stack.h X C stack.cpp C main.cpp

> LAPRAK MODUL 7

> OUTLINE

stack.h

```
1 #include <iostream>
2 #include "stack.h"
3 using namespace std;
4
5 void createStack(Stack &S) {
6     S.Top = 0;
7 }
8 bool isEmpty(Stack S) {
9     return S.Top == 0;
10 }
11 bool isFull(Stack S) {
12     return S.Top == 20;
13 }
14
15 void push(Stack &S, infotype x) {
16     if (!isFull(S)) {
17         S.Top++;
18         S.info[S.Top] = x;
19     } else {
20         cout << "Stack penuh!" << endl;
21     }
22 }
23
24 infotype pop(Stack &S) {
25     infotype x = 0;
26     if (!isEmpty(S)) {
```

nama & nim - ...

File Edit Format View Help

Nama : Muhammad Azzam Satria

NIM : 103112400112

1f Windows (CRLF) UTF-8

The screenshots show two separate code editor sessions, likely from Microsoft Visual Studio Code, demonstrating the execution of C++ programs related to a stack data structure.

Top Window (Screenshot 1):

```

27     |     x = S.info[S.Top];
28     |     S.Top--;
29 } else {
30     |     cout << "Stack kosong!" << endl;
31 }
32     |     return x;
33 }

34 void printInfo(Stack S) {
35     if (isEmpty(S)) {
36         |     cout << "Stack kosong." << endl;
37     } else {
38         for (int i = S.Top; i >= 1; i--) {
39             |             cout << S.info[i] << " ";
40         }
41         cout << endl;
42     }
43 }

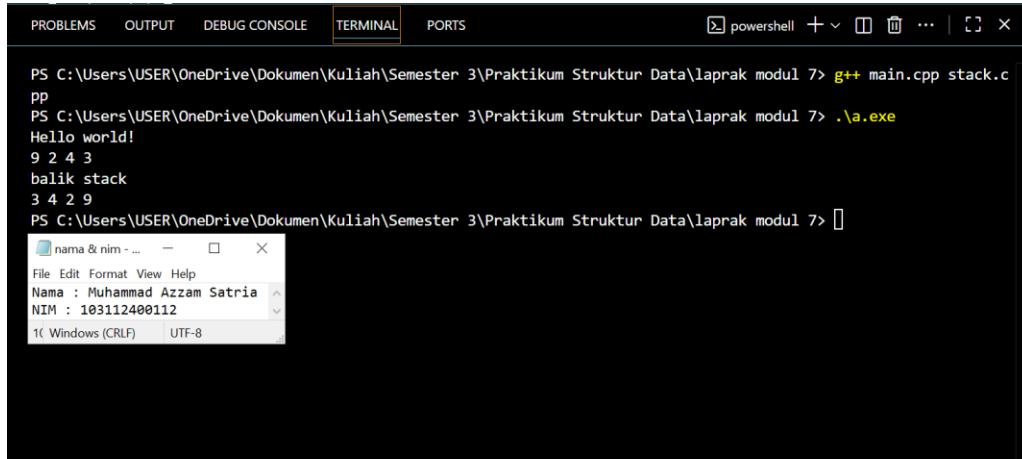
44 void balikStack(Stack &S) {
45     Stack temp;
46     createStack(temp);
47     while (!isEmpty(S)) {
48         |         push(temp, pop(S));
49     }
50     S = temp;
51 }
52 }
```

Bottom Window (Screenshot 2):

```

1 #include <iostream>
2 #include "stack.h"
3 using namespace std;
4
5 int main() {
6     cout << "Hello world!" << endl;
7     Stack S;
8
9     createStack(S);
10    push(S, 3);
11    push(S, 4);
12    push(S, 8);
13    pop(S);
14    push(S, 2);
15    push(S, 3);
16    pop(S);
17    push(S, 9);
18
19    printInfo(S);
20    cout << "balik stack" << endl;
21    balikStack(S);
22    printInfo(S);
23    return 0;
24 }
```

Screenshots Output



```
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak modul 7> g++ main.cpp stack.c
PP
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak modul 7> .\a.exe
Hello world!
balik stack
3 4 2 9
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak modul 7> 
```

Deskripsi:

Program ini merupakan struktur data stack dengan menggunakan array berukuran 20 elemen. Di dalam program terdapat komponen utama yaitu “info” sebagai penyimpan data dengan tipe integer dan “top” sebagai penanda posisi elemen teratas. Fungsi createStack yaitu untuk menginisialisasi stack agar kosong dengan mengatur nilai top menjadi 0. Fungsi push yaitu untuk menambahkan data ke posisi teratas jika stack belum penuh. Fungsi pop yaitu untuk menghapus elemen teratas apabila stack tidak kosong. Fungsi printInfo yaitu untuk menampilkan isi stack dari elemen paling atas hingga bawah.

Selain itu, terdapat prosedur balikStack untuk membalik urutan isi stack dengan memanfaatkan stack sementara. Fungsi isEmpty dan isFull yaitu untuk mengecek apakah kondisi stack sedang kosong atau penuh sebelum program dijalankan. Pada fungsi main terdapat operasi push dan pop untuk menguji Semua fungsi stack yang dibuat. Output akan menampilkan kondisi stack sebelum dan sesudah proses pembalikan.

Unguided 2 Penambahan prosedur pushAscending

The screenshot shows the Visual Studio Code interface with two open files:

- stack.h**: Contains the header file for the stack structure.
- stack.cpp**: Contains the implementation of the stack class, including the `pushAscending` method which handles pushing elements onto a stack.
- main.cpp**: Contains the main program logic, including creating a stack, pushing elements, and printing the stack's state.

A terminal window at the bottom displays the following output:

```

PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak modul 7> g++ main.cpp stack.c
PP
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak modul 7> .\a.exe
Hello world!
9 8 4 3 3 2
balik stack
2 3 3 4 8 9
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak modul 7> []
  
```

Screenshot output

The terminal window in VS Code shows the following command and output:

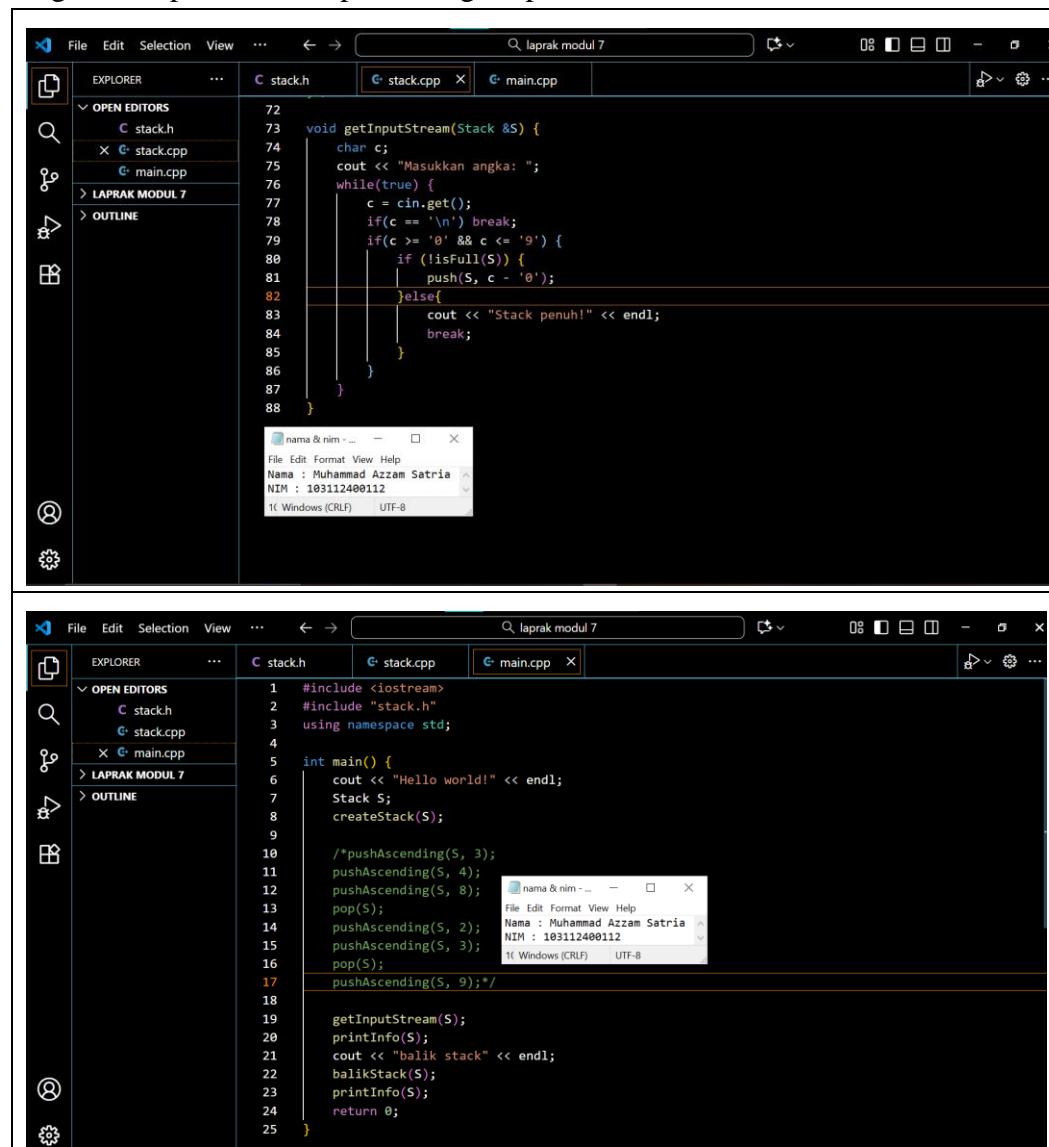
```

PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak modul 7> g++ main.cpp stack.c
PP
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak modul 7> .\a.exe
Hello world!
9 8 4 3 3 2
balik stack
2 3 3 4 8 9
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak modul 7> []
  
```

Deskripsi

Pada soal nomor 2 terdapat penambahan prosedur pushAscending yang fungsinya yaitu untuk menambahkan data ke dalam stack secara berurutan dari nilai yang kecil ke nilai yang besar. Prosedur ini memakai satu stack tambahan untuk menampung sementara data yang lebih besar dari angka yang akan dimasukkan. Kemudian setelah menemukan posisi yang tepat, data baru tadi akan dimasukkan ke stack utama, lalu data sementara dikembalikan seperti semula.

Unguided 3 penambahan prosedur getInputStream



```
File Edit Selection View ... ← → Q laprak modul 7
EXPLORER ... C stack.h E stack.cpp X C main.cpp ...
OPEN EDITORS
C stack.h
X E stack.cpp
C main.cpp
LAPRAK MODUL 7
OUTLINE
72
73 void getInputStream(Stack &S) {
74     char c;
75     cout << "Masukkan angka: ";
76     while(true) {
77         c = cin.get();
78         if(c == '\n') break;
79         if(c >= '0' && c <= '9') {
80             if (!isFull(S)) {
81                 push(S, c - '0');
82             } else{
83                 cout << "Stack penuh!" << endl;
84                 break;
85             }
86         }
87     }
88 }
```

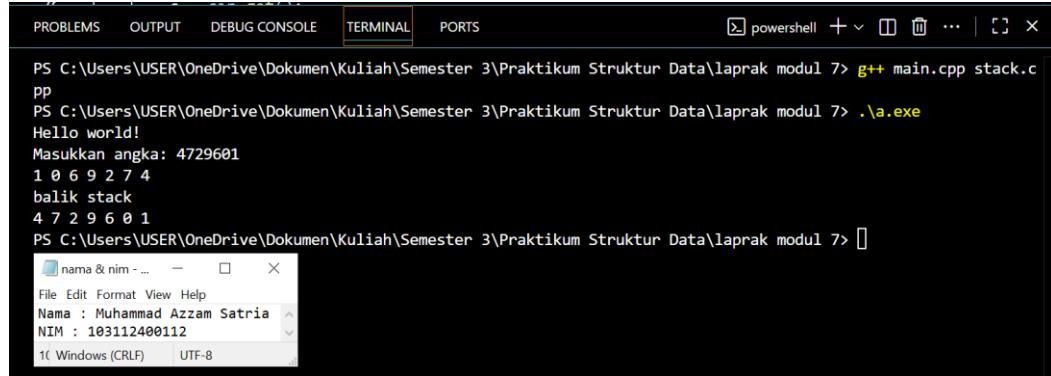
```
File Edit Format View Help
Nama : Muhammad Azzam Satria
NIM : 103112400112
1 Windows (CRLF) UTF-8
```



```
File Edit Selection View ... ← → Q laprak modul 7
EXPLORER ... C stack.h E stack.cpp X C main.cpp ...
OPEN EDITORS
C stack.h
X E stack.cpp
C main.cpp
LAPRAK MODUL 7
OUTLINE
1 #include <iostream>
2 #include "stack.h"
3 using namespace std;
4
5 int main() {
6     cout << "Hello world!" << endl;
7     Stack S;
8     createStack(S);
9
10    /*pushAscending(S, 3);
11    pushAscending(S, 4);
12    pushAscending(S, 8);
13    pop(S);
14    pushAscending(S, 2);
15    pushAscending(S, 3);
16    pop(S);
17    pushAscending(S, 9);*/
18
19    getInputStream(S);
20    printInfo(S);
21    cout << "balik stack" << endl;
22    balikStack(S);
23    printInfo(S);
24    return 0;
25 }
```

```
File Edit Format View Help
Nama : Muhammad Azzam Satria
NIM : 103112400112
1 Windows (CRLF) UTF-8
```

Screenshot output



```
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak modul 7> g++ main.cpp stack.c
PP
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak modul 7> .\a.exe
Hello world!
Masukkan angka: 4729601
1 0 6 9 2 7 4
balik stack
4 7 2 9 6 0 1
PS C:\Users\USER\OneDrive\Dokumen\Kuliah\Semester 3\Praktikum Struktur Data\laprak modul 7> []
```

nama & nim - ... — □ ×
File Edit Format View Help
Nama : Muhammad Azzam Satria
NIM : 183112400112
10 Windows (CRLF) UTF-8

Deskripsi

Pada soal nomor 3 terdapat penambahan prosedur getInputStream yang fungsinya yaitu untuk membaca angka yang diinput oleh user. Setiap angka yang diinputkan akan diubah dari bentuk karakter menjadi nilai angka sebelum dimasukkan ke dalam stack. Kemudian data akan disimpan dengan nilai angka sebenarnya. Lalu isi stack akan ditampilkan atau dibalik dengan menggunakan prosedur balikStack.

D. Kesimpulan

Stack merupakan salah satu struktur data yang menyimpan data dengan cara data yang terakhir dimasukkan, maka akan menjadi data yang pertama dikeluarkan. Dengan adanya contoh dan latihan pada modul, saya belajar secara langsung cara membuat dan mengelola stack menggunakan array dengan beberapa operasi dasar yaitu ada push, pop, printInfo dan balikStack. Operasi dasar tadi memiliki fungsi masing-masing seperti push untuk menambahkan data, pop untuk menghapus data dan balikStack untuk membalikkan urutan isi stack.

Selain operasi dasar tadi, pada soal latihan terdapat prosedur tambahan yaitu pushAscending dan getInputStream. Fungsi dari prosedur pushAscending yaitu untuk membantu menjaga agar data tetap tersusun secara berurutan saat dimasukkan, sedangkan prosedur getInputStream untuk menginputkan data ke dalam stack secara langsung tanpa harus menambahkannya pada kode program, sehingga data dapat dimasukkan sesuai kebutuhan saat program dijalankan.

E. Referensi

- Setiyawan, R. D., Hermawan, D., Abdillah, A. F., Mujayanah, A., & Vindua, R. (2024). *Penggunaan struktur data stack dalam pemrograman C++ dengan pendekatan array dan linked list*. *JUTECH: Journal Education and Technology*, 5(2), 484–498.
- Ginting, S. H. N., Effendi, H., Kumar, S., Marsisno, W., Sitanggang, Y. R. U., Anwar, K., & Smrti, N. N. E. (2024). *Pengantar struktur data*. Penerbit Mifandi Mandiri Digital, 1(1).