

**LAPORAN PRAKTIKUM  
STRUKTUR DATA**

**MODUL VII**

**STACK**



**Disusun Oleh :**

NAMA : Muhammad Fachri Auravyano Saka  
NIM : 103112430180

**Dosen**

FAHRUDIN MUKTI WIBOWO

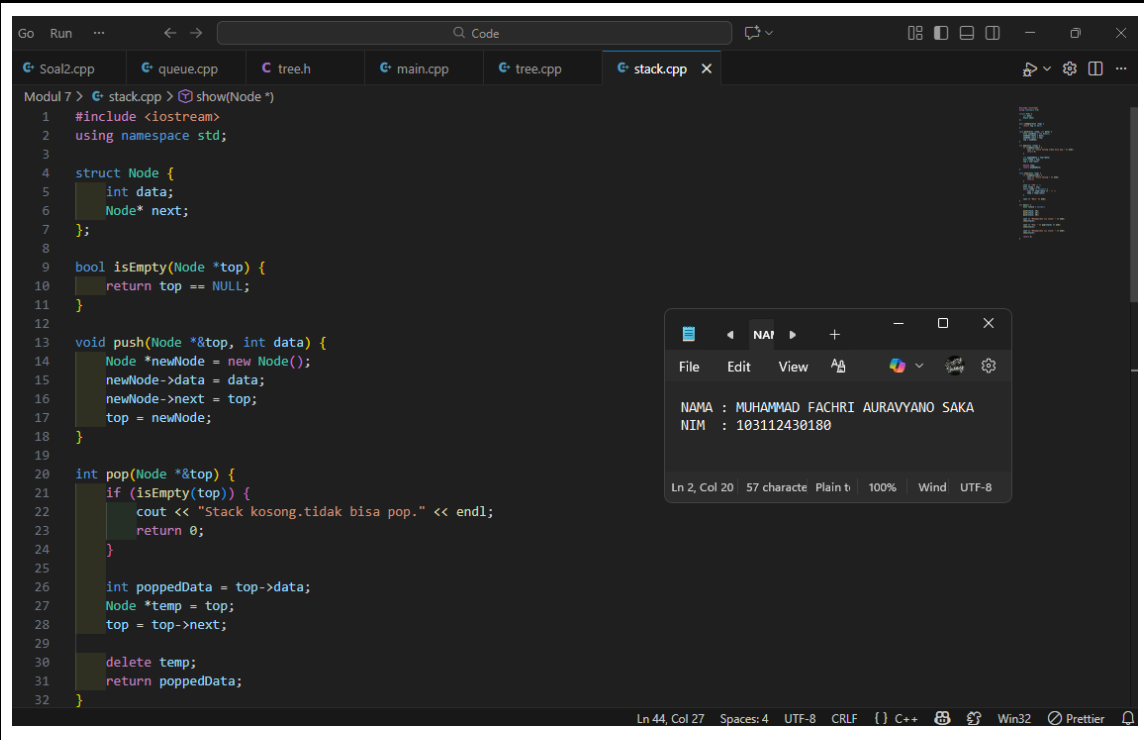
**PROGRAM STUDI STRUKTUR DATA  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY PURWOKERTO  
2025**

## A. Dasar Teori

Stack atau tumpukan adalah struktur data linier yang bekerja dengan prinsip LIFO (Last In, First Out). Artinya, data yang terakhir kali dimasukkan akan menjadi data yang pertama kali diambil, mirip seperti tumpukan piring di mana kita hanya bisa mengambil atau menambah piring dari posisi paling atas. Dalam dunia pemrograman, Stack sangat berguna untuk memproses data yang perlu dibalik urutannya atau menyimpan riwayat aktivitas (seperti fitur undo). Dalam implementasi menggunakan tabel atau Array, Stack memiliki kapasitas maksimum yang tetap (MAXSTACK). Komponen terpenting dalam Stack adalah TOP, yaitu penanda indeks elemen paling atas. Operasi utama pada Stack ada dua: Push untuk menambahkan data baru ke posisi TOP (indeks bertambah), dan Pop untuk mengambil data dari posisi TOP (indeks berkurang). Selain itu, terdapat operasi createStack untuk inisialisasi awal dan pengecekan kondisi seperti isEmpty (kosong) atau isFull (penuh) untuk mencegah kesalahan akses data.

## B. Guided (berisi screenshot source code & output program disertai penjelasannya)

### Guided 1



The screenshot displays a C++ IDE with a file explorer at the top showing several files: Soal2.cpp, queue.cpp, tree.h, main.cpp, tree.cpp, and stack.cpp. The main editor window shows the source code for stack.cpp. The code defines a Node struct with an integer data field and a pointer to the next Node. It implements isEmpty, push, and pop functions. The push function adds a new node to the top of the stack, and the pop function removes the top node. The output window on the right shows the program's output: NAMA : MUHAMMAD FACHRI AURAVYANO SAKA and NIM : 103112430180. The status bar at the bottom indicates the current line and column (Ln 44, Col 27), the number of spaces (4), the encoding (UTF-8), the line ending (CRLF), the language (C++), the window title (Win32), and the formatter (Prettier).

```
Modul 7 > stack.cpp > show(Node *)
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* next;
7  };
8
9  bool isEmpty(Node *top) {
10     return top == NULL;
11 }
12
13 void push(Node *&top, int data) {
14     Node *newNode = new Node();
15     newNode->data = data;
16     newNode->next = top;
17     top = newNode;
18 }
19
20 int pop(Node *&top) {
21     if (isEmpty(top)) {
22         cout << "Stack kosong.tidak bisa pop." << endl;
23         return 0;
24     }
25
26     int poppedData = top->data;
27     Node *temp = top;
28     top = top->next;
29
30     delete temp;
31     return poppedData;
32 }
```

NAMA : MUHAMMAD FACHRI AURAVYANO SAKA  
NIM : 103112430180

Ln 44, Col 27 Spaces: 4 UTF-8 CRLF {} C++ Win32 Prettier

```
Modul 7 > stack.cpp > show(Node *)
34 void show(Node *top) {
35     if (isEmpty(top)) {
36         cout << "Stack kosong." << endl;
37         return;
38     }
39     cout << "TOP -> ";
40     Node *temp = top;
41     while (temp != nullptr) {
42         cout << temp->data << " -> ";
43         temp = temp->next;
44     }
45     cout << "NULL" << endl;
46 }
47
48 int main() {
49     Node *stack = nullptr;
50     push(stack, 10);
51     push(stack, 20);
52     push(stack, 30);
53     cout << "Menampilkan isi stack: " << endl;
54     show(stack);
55     cout << "Pop: " << pop(stack) << endl;
56     show(stack);
57     cout << "Menampilkan isi stack: " << endl;
58     show(stack);
59 }
60
61 int main() {
62     Node *stack = nullptr;
63     push(stack, 10);
64     push(stack, 20);
65     push(stack, 30);
66     cout << "Menampilkan isi stack: " << endl;
67     show(stack);
68     cout << "Pop: " << pop(stack) << endl;
69     show(stack);
70     cout << "Menampilkan isi stack: " << endl;
71     show(stack);
72     return 0;
73 }
```

## Screenshots Output

```
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code\Modul 7> cd "e:\KULIAH\SEMESTER 3\Struktur Data\Code\Modul 7\" ; if ($?) { g++ stack.cpp -o stack }; if ($?) { .\stack }
Menampilkan isi stack:
TOP -> 30 -> 20 -> 10 -> NULL
Pop: 30
TOP -> 20 -> 10 -> NULL
Menampilkan isi stack:
TOP -> 20 -> 10 -> NULL
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code\Modul 7>
```

## Deskripsi:

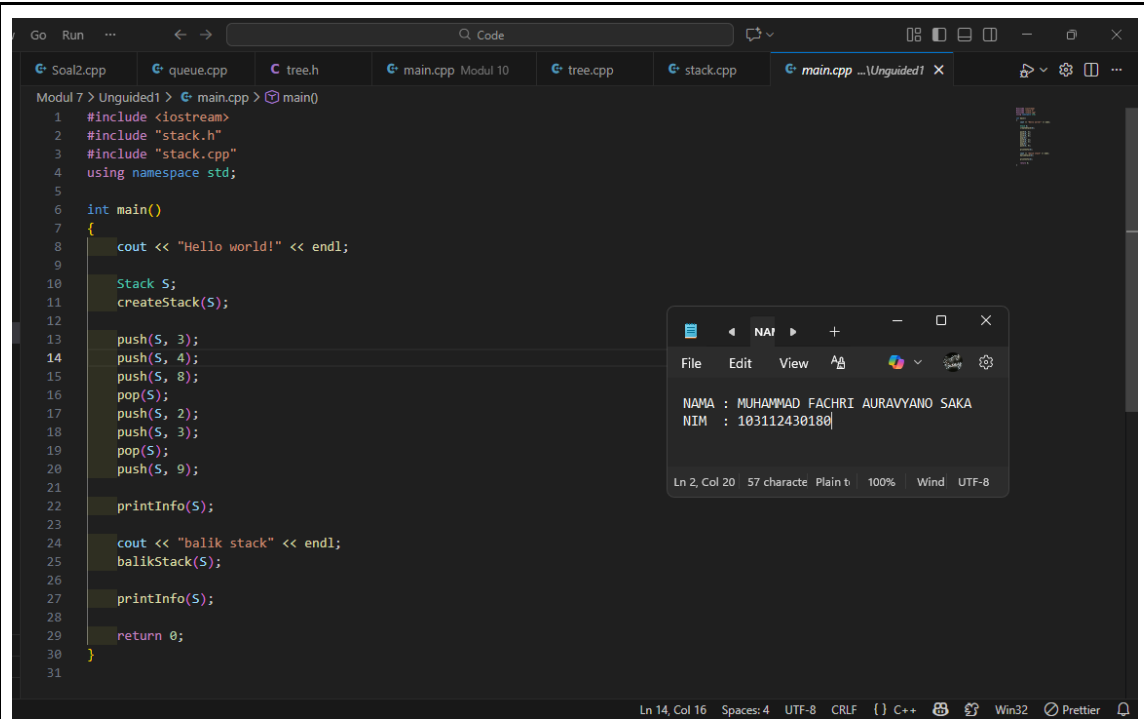
Program C++ tersebut adalah implementasi sederhana dari struktur data stack menggunakan linked list. Stack bekerja dengan konsep LIFO (Last In, First Out), artinya

data yang terakhir dimasukkan akan menjadi data yang pertama diambil. Program ini memiliki beberapa fungsi utama: `push()` untuk menambahkan data ke bagian atas stack, `pop()` untuk menghapus dan mengembalikan data teratas, serta `show()` untuk menampilkan seluruh isi stack dari atas ke bawah. Struktur Node digunakan untuk menyimpan data dan pointer ke node berikutnya. Di dalam fungsi `main`, program menambahkan tiga data (10, 20, 30), menampilkannya, kemudian melakukan `pop` satu kali dan menampilkan isi stack kembali. Program ini membantu memahami cara kerja stack secara dinamis menggunakan pointer.

C. Unguided/Tugas (berisi screenshot source code & output program disertai penjelasannya)

Unguided 1

Main.cpp



```
Modul 7 > Unguided1 > main.cpp > main()
1  #include <iostream>
2  #include "stack.h"
3  #include "stack.cpp"
4  using namespace std;
5
6  int main()
7  {
8      cout << "Hello world!" << endl;
9
10     Stack S;
11     createStack(S);
12
13     push(S, 3);
14     push(S, 4);
15     push(S, 8);
16     pop(S);
17     push(S, 2);
18     push(S, 3);
19     pop(S);
20     push(S, 9);
21
22     printInfo(S);
23
24     cout << "balik stack" << endl;
25     balikStack(S);
26
27     printInfo(S);
28
29     return 0;
30 }
31
```

File Edit View A 100% Wind UTF-8

NAMA : MUHAMMAD FACHRI AURAVYANO SAKA  
NIM : 103112430180

Ln 2, Col 20 57 caracte Plain t 100% Wind UTF-8

Ln 14, Col 16 Spaces: 4 UTF-8 CRLF C++ Win32 Prettier

Stack.cpp

```
Go Run ... < > Code
Soal2.cpp queue.cpp tree.h main.cpp tree.cpp stack.cpp Modul 7 stack.cpp ...\Unguided1 X
Modul 7 > Unguided1 > stack.cpp > ...
1 #include <iostream>
2 #include "stack.h"
3 using namespace std;
4
5 void createStack(Stack &S) {
6     S.top = -1; // Stack kosong
7 }
8
9 void push(Stack &S, infotype x) {
10     if (S.top < MAXSTACK - 1) {
11         S.top++;
12         S.info[S.top] = x;
13     } else {
14         cout << "Stack penuh!" << endl;
15     }
16 }
17
18 infotype pop(Stack &S) {
19     if (S.top >= 0) {
20         infotype x = S.info[S.top];
21         S.top--;
22         return x;
23     } else {
24         cout << "Stack kosong!" << endl;
25         return -1;
26     }
27 }
28
29 void printInfo(Stack S) {
30     if (S.top < 0) {
31         cout << "[STACK KOSONG]" << endl;
32     } else {
```

```
Go Run ... < > Code
Soal2.cpp queue.cpp tree.h main.cpp tree.cpp stack.cpp Modul 7 stack.cpp ...\Unguided1 X
Modul 7 > Unguided1 > stack.cpp > ...
18 infotype pop(Stack &S) {
19 }
20
21 void printInfo(Stack S) {
22     if (S.top < 0) {
23         cout << "[STACK KOSONG]" << endl;
24     } else {
25         cout << "[TOP] ";
26         for (int i = S.top; i >= 0; i--) {
27             cout << S.info[i] << " ";
28         }
29         cout << endl;
30     }
31 }
32
33 void balikStack(Stack &S) {
34     Stack temp;
35     createStack(temp);
36
37     while (S.top >= 0) {
38         push(temp, pop(S));
39     }
40
41     S = temp; // Copy stack balik
42 }
43
44
45
46
47
48
49
50
51
```

Stack.h

```
1 #ifndef STACK_H
2 #define STACK_H
3
4 const int MAXSTACK = 20;
5
6 typedef int infotype;
7
8 struct Stack {
9     infotype info[MAXSTACK];
10     int top;
11 };
12
13 // PROTOTYPE
14 void createStack(Stack &S);
15 void push(Stack &S, infotype x);
16 infotype pop(Stack &S);
17 void printInfo(Stack S);
18 void balikStack(Stack &S);
19
20 #endif
21
```

NAMA : MUHAMMAD FACHRI AURAVYANO SAKA  
NIM : 103112430180

## Screenshot Output

```
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code\Modul 7> cd "e:\KULIAH\SEMESTER 3\Struktur Data\Code\Modul 7" & gcc main.cpp & ./main
) { .\main }
Hello world!
[TOP] 9 2 4 3
balik stack
[TOP] 3 4 2 9
PS E:\KULIAH\SEMESTER 3\Struktur Data\Code\Modul 7\Unguided1>
```

NAMA : MUHAMMAD FACHRI AURAVYANO SAKA  
NIM : 103112430180

## Deskripsi:

Program C++ ini mensimulasikan Stack (tumpukan) angka sederhana yang bekerja dengan prinsip "terakhir masuk, pertama keluar" (LIFO), mirip seperti tumpukan piring di mana anda hanya bisa mengambil atau menambah piring dari posisi paling atas. Di dalam fungsi utama (main), program melakukan serangkaian operasi campuran: memasukkan angka (push), membuang angka teratas (pop), dan menambahkan angka baru lagi hingga tersisa tumpukan akhir [3, 4, 2, 9]. Setelah menampilkan isi tumpukan tersebut, program menggunakan fungsi khusus untuk membalik urutan seluruh angka di dalamnya—menjadikan angka yang tadinya paling bawah menjadi paling atas—dan menampilkan hasil akhirnya ke layar.

## Unguided 2

```
Go Run ... < -> Code
Soal2.cpp queue.cpp tree.h main.cpp tree.cpp stack.cpp unguided2.cpp x
Modul 7 > Unguided2 > unguided2.cpp > createStack(Stack &)
1 #include <iostream>
2 using namespace std;
3
4 const int MAX = 20;
5
6 struct Stack {
7     int info[MAX];
8     int Top;
9 };
10
11 void createStack(Stack &S){
12     S.Top = 0;
13 }
14
15 bool isEmpty(Stack S){
16     return (S.Top == 0);
17 }
18
19 bool isFull(Stack S){
20     return (S.Top == MAX);
21 }
22
23 void push(Stack &S, int x){
24     if(!isFull(S)){
25         S.info[S.Top] = x;
26         S.Top++;
27     }
28 }
29
30 int pop(Stack &S){
31     if(!isEmpty(S)){
32         S.Top--;
```

File Edit View A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9

NAMA : MUHAMMAD FACHRI AURAVYANO SAKA  
NIM : 103112430180

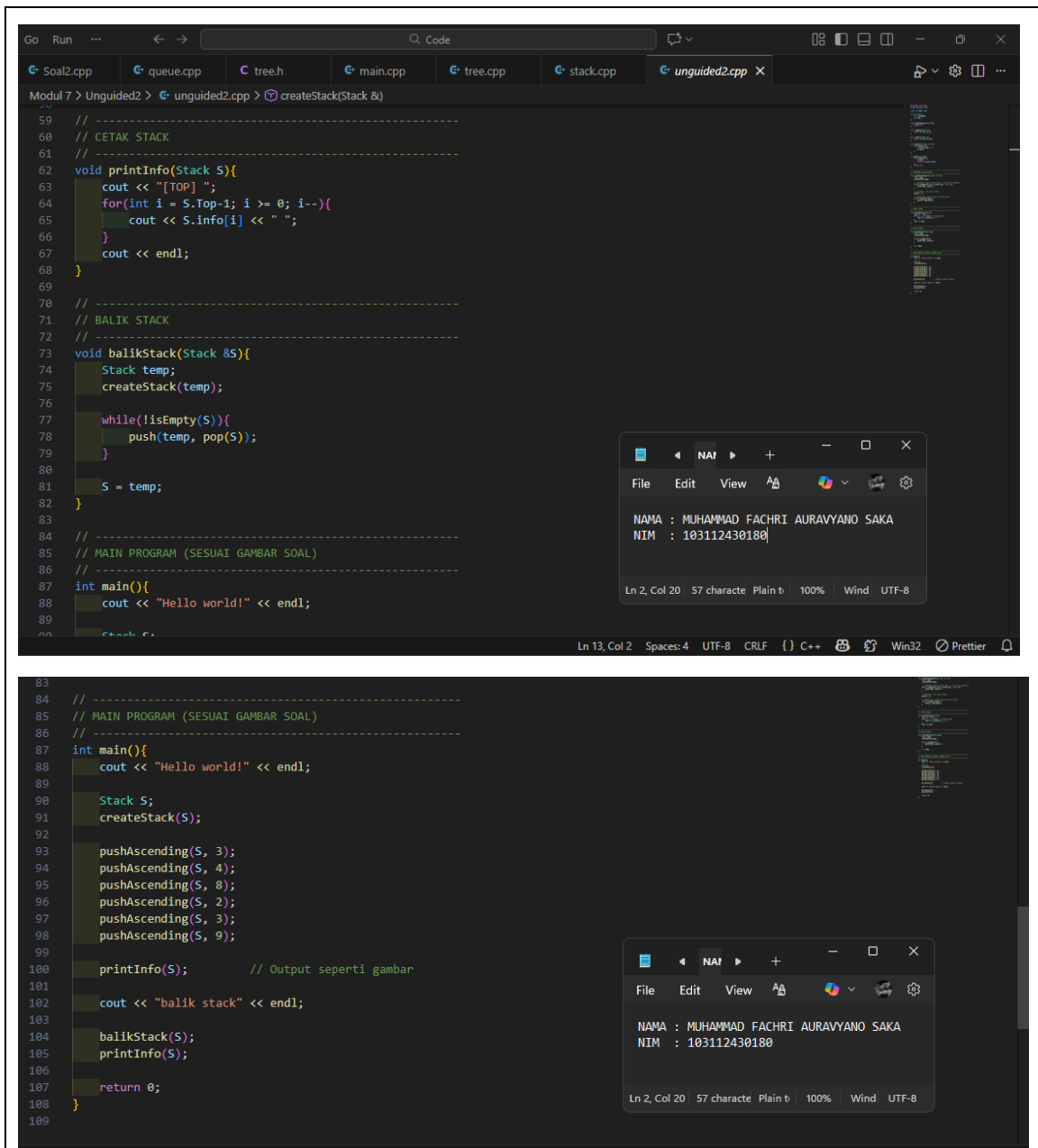
Ln 2, Col 20 57 character Plain text 100% Wind UTF-8

```
Go Run ... < -> Code
Soal2.cpp queue.cpp tree.h main.cpp tree.cpp stack.cpp unguided2.cpp x
Modul 7 > Unguided2 > unguided2.cpp > createStack(Stack &)
29
30 int pop(Stack &S){
31     if(!isEmpty(S)){
32         S.Top--;
33         return S.info[S.Top];
34     }
35     return -1;
36 }
37
38 // -----
39 // PROSEDUR pushAscending
40 // -----
41 void pushAscending(Stack &S, int x){
42     Stack temp;
43     createStack(temp);
44
45     // pindahkan semua elemen yang > x ke stack sementara
46     while(!isEmpty(S) && S.info[S.Top - 1] > x){
47         push(temp, pop(S));
48     }
49
50     // masukkan x ke stack utama
51     push(S, x);
52
53     // kembalikan elemen temp ke stack utama
54     while(!isEmpty(temp)){
55         push(S, pop(temp));
56     }
57 }
58
59 // -----
60 // CETAK STACK
```

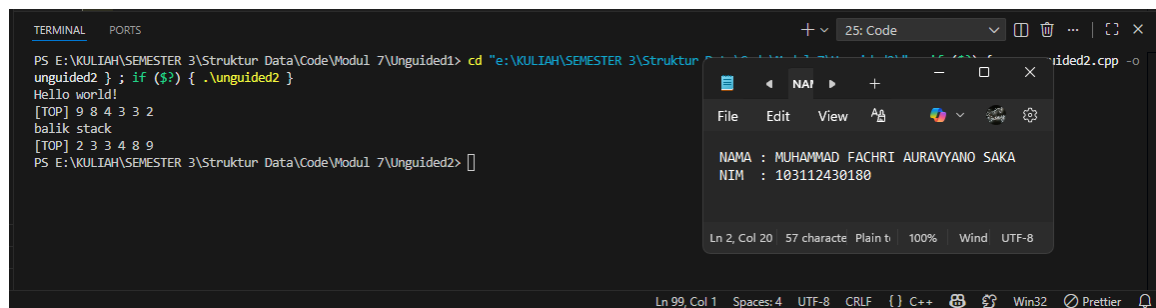
File Edit View A B C D E F G H I J K L M N O P Q R S T U V W X Y Z 0 1 2 3 4 5 6 7 8 9

NAMA : MUHAMMAD FACHRI AURAVYANO SAKA  
NIM : 103112430180

Ln 2, Col 20 57 character Plain text 100% Wind UTF-8



## Screenshot Output



## Deskripsi:

Program ini merupakan implementasi struktur data stack menggunakan array berukuran tetap. Stack bekerja dengan prinsip LIFO (Last In, First Out). Program menyediakan operasi dasar seperti membuat stack (createStack), mengecek apakah stack kosong atau



peny, memasukkan data (push), dan menghapus data (pop). Selain itu, terdapat fungsi khusus bernama pushAscending yang akan memasukkan elemen baru ke dalam stack dengan urutan menaik (ascending), sehingga stack selalu tersusun dari nilai kecil di bawah ke nilai besar di atas. Ada juga fungsi balikStack yang membalik urutan elemen stack. Pada bagian main, beberapa angka dimasukkan menggunakan pushAscending, kemudian program menampilkan isi stack dan membaliknya. Program ini bertujuan untuk menunjukkan cara mengelola stack sekaligus menjaga urutan data menggunakan stack tambahan.

### Unguided 3

```
Modul 7 > Unguided3 > unguided3.cpp > getInputStream(Stack &S)
1  #include <iostream>
2  using namespace std;
3
4  const int MAX = 20;
5
6  struct Stack {
7      int info[MAX];
8      int Top;
9  };
10
11 void createStack(Stack &S){
12     S.Top = 0;
13 }
14
15 bool isEmpty(Stack S){
16     return (S.Top == 0);
17 }
18
19 bool isFull(Stack S){
20     return (S.Top == MAX);
21 }
22
23 void push(Stack &S, int x){
24     if(!isFull(S)){
25         S.info[S.Top] = x;
26         S.Top++;
27     }
28 }
29
30 int pop(Stack &S){
31     if(!isEmpty(S)){
32         S.Top--;
33         return S.info[S.Top];
34     }
35     return -1;
36 }
37
38 // -----
39 // PROSEDUR getInputStream
40 // Membaca input karakter sampai user menekan ENTER
41 // -----
42 void getInputStream(Stack &S){
43     cout << "Masukkan input : ";
44
45     char c;
46     while (true){
47         c = cin.get(); // baca 1 karakter
48
49         if (c == '\n') // jika ENTER -> berhenti
50             break;
51
52         int angka = c - '0'; // ubah char ke int
53         push(S, angka);
54     }
55 }
56
57 // -----
58 void printInfo(Stack S){
59     cout << "[TOP] ";
60     for(int i = S.Top-1; i >= 0; i--){
```

File Edit View

NAMA : MUHAMMAD FACHRI AURAVYANO SAKA  
NIM : 103112430180

Ln 2, Col 20 57 character Plain t 100% Wind UTF-8

```
29
30 int pop(Stack &S){
31     if(!isEmpty(S)){
32         S.Top--;
33         return S.info[S.Top];
34     }
35     return -1;
36 }
37
38 // -----
39 // PROSEDUR getInputStream
40 // Membaca input karakter sampai user menekan ENTER
41 // -----
42 void getInputStream(Stack &S){
43     cout << "Masukkan input : ";
44
45     char c;
46     while (true){
47         c = cin.get(); // baca 1 karakter
48
49         if (c == '\n') // jika ENTER -> berhenti
50             break;
51
52         int angka = c - '0'; // ubah char ke int
53         push(S, angka);
54     }
55 }
56
57 // -----
58 void printInfo(Stack S){
59     cout << "[TOP] ";
60     for(int i = S.Top-1; i >= 0; i--){
```

File Edit View

NAMA : MUHAMMAD FACHRI AURAVYANO SAKA  
NIM : 103112430180

Ln 2, Col 20 57 character Plain t 100% Wind UTF-8

```
Ln 43, Col 30 Spaces: 4 UTF-8 CRLF {} C++ Win32 Prettier
```

```
57 // -----
58 void printInfo(Stack S){
59     cout << "[TOP] ";
60     for(int i = S.Top-1; i >= 0; i--){
61         cout << S.info[i] << " ";
62     }
63     cout << endl;
64 }
65 // -----
66 void balikStack(Stack &S){
67     Stack temp;
68     createStack(temp);
69     while(!isEmpty(S)){
70         push(temp, pop(S));
71     }
72     S = temp;
73 }
74 // -----
75 int main(){
76     cout << "Hello world!" << endl;
77     Stack S;
78     createStack(S);
79     getInputStream(S);
80     printInfo(S);
81     cout << "balik stack" << endl;
82     balikStack(S);
83     printInfo(S);
84     return 0;
85 }
```

Terminal Output:

```
NAMA : MUHAMMAD FACHRI AURAVYANO SAKA
NIM : 103112430180
```

#### Deskripsi:

Program ini adalah implementasi stack menggunakan array yang berfungsi untuk membaca input berupa deretan karakter angka yang dimasukkan pengguna hingga menekan tombol ENTER. Setiap karakter angka langsung diubah menjadi bilangan integer dan disimpan ke dalam stack menggunakan operasi push. Karena stack bekerja dengan prinsip LIFO (Last In, First Out), angka yang terakhir dimasukkan akan berada di posisi paling atas. Program juga memiliki fitur untuk menampilkan isi stack dari atas ke bawah, serta fungsi balikStack yang membalik urutan semua elemen menggunakan stack sementara. Pada bagian main, program meminta input dari pengguna, menampilkan isi stack, lalu membalik stack dan menampilkannya kembali. Dengan demikian, program ini membantu memperlihatkan cara kerja stack dalam membaca dan memproses input secara terbalik.

#### D. Kesimpulan

Berdasarkan praktikum yang telah dilakukan, dapat disimpulkan bahwa struktur data Stack berhasil diimplementasikan menggunakan Array dalam bahasa C++. Prinsip LIFO terbukti berjalan dengan baik, di mana pengelolaan variabel TOP menjadi kunci utama; TOP akan naik saat data di-push dan turun saat data di-pop. Penggunaan array

membuat akses data menjadi cepat, namun jumlah data yang disimpan terbatas sesuai ukuran array yang didefinisikan di awal.

Melalui tugas yang dikerjakan, penerapan Stack terbukti efektif untuk menyelesaikan berbagai masalah logika. Pada program Unguided, Stack digunakan untuk membalikkan urutan angka (`balikStack`), mengurutkan data secara otomatis saat input (`pushAscending`), hingga menangani aliran input karakter (`getInputStream`) sampai tombol Enter ditekan. Hal ini menunjukkan fleksibilitas Stack dalam memanipulasi urutan data secara efisien.

#### 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.
- Korostinskiy, R., Darashkevich, E., Rusyaev, R., & Bugayenko, Y. (2024). Heap vs. Stack: Analyzing Memory Allocations in C and C++ Open Source Software. *arXiv preprint arXiv:2403.06695*.
- Schmidt, T. C. Redesigning and Evaluating the Network Stack in the C++ Actor Framework.