

**LAPORAN PRAKTIKUM
STRUKTUR DATA**

**MODUL I
PENGENALAN CODE BLOCKS**



Disusun Oleh :
NAMA : Jauza Rasyiq Hernanta
NIM : **103112430033**

Dosen
WAHYU ANDI SAPUTRA

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

A. Dasar Teori

Stack merupakan salah satu bentuk struktur data dimana prinsip operasi yang digunakan seperti tumpukan. Seperti halnya tumpukan, elemen yang bisa diambil terlebih dahulu adalah elemen yang paling atas, atau elemen yang pertama kali masuk, prinsip ini biasa disebut LIFO (Last In First Out).

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

Guided 1

Main.cpp

```
Modul 6 > C main.cpp > ...
1 #include "kendaraan.h"
2 #include "kendaraan.cpp"
3 #include <iostream>
4 #include <limits>
5
6 using namespace std;
7
8 void clearInputBuffer() {
9     cin.clear();
10    cin.ignore(numeric_limits<streamsize>::max(), '\n');
11 }
12
13 void displayMenu() {
14     cout << "\n==== MENU MANAJEMEN KENDARAAN ===\n";
15     cout << "1. Tambah Data Kendaraan\n";
16     cout << "2. Tampilkan Semua Data\n";
17     cout << "3. Keluar\n";
18     cout << "Pilihan: ";
19 }
20
21 Kendaraan inputKendaraan() {
22     Kendaraan k;
23
24     cout << "\n--- Input Data Kendaraan ---\n";
25
26     cout << "Masukkan nomor polisi: ";
27     getline(cin, k.nopol);
28
29     cout << "Masukkan warna kendaraan: ";
30     getline(cin, k.warna);
31
32     cout << "Masukkan tahun kendaraan: ";
33     while (!(cin >> k.thnBuat)) {
34         cout << "Input tahun harus angka! Coba lagi: ";
35         cin.clear();
36         clearInputBuffer();
37     }
38     clearInputBuffer();
39
40     return k;
41 }
42
```

```
42 int main() {
43     List L;
44     CreateList(L);
45     int choice;
46
47     do {
48         displayMenu();
49         cin >> choice;
50         clearInputBuffer();
51
52         switch (choice) {
53             case 1: {
54                 Kendaraan newKendaraan = inputKendaraan();
55
56                 if (isNopolExist(L, newKendaraan.nopol)) {
57                     cout << "\nNomor polisi '" << newKendaraan.nopol
58                     << "' sudah terdaftar!\n";
59                 } else {
60                     address newNode = alokasi(newKendaraan);
61                     insertLast(L, newNode);
62                     cout << "\nData kendaraan berhasil ditambahkan!\n";
63                 }
64             }
65             break;
66         }
67
68         case 2:
69             printInfo(L);
70             break;
71
72         case 3:
73             cout << "\nProgram selesai.\n";
74             break;
75
76         default:
77             cout << "\nPilihan tidak valid!\n";
78         }
79     } while (choice != 3);
80
81     address current = L.First;
82     while (current != nullptr) {
83         address nextNode = current->next;
84         dealokasi(current);
85         current = nextNode;
86     }
87 }
88
89     return 0;
90 }
```

Kendaraan.cpp

```

4     using namespace std;
5
6 void CreateList(List &L) {
7     L.First = nullptr;
8     L.Last = nullptr;
9 }
10
11 address alokasi(Kendaraan x) {
12     address P = new EIMList;
13     P->info = x;
14     P->next = nullptr;
15     P->prev = nullptr;
16     return P;
17 }
18
19 void dealokasi(address &P) {
20     delete P;
21     P = nullptr;
22 }
23
24 void printInfo(List L) {
25     if (L.First == nullptr) {
26         cout << "\nDATA LIST KOSONG\n";
27         return;
28     }
29
30     cout << "\nDATA LIST\n";
31     cout << "=====\\n";
32
33     address P = L.First;
34     int count = 1;
35
36     while (P != nullptr) {
37         cout << "Kendaraan #" << count << ":\n";
38         cout << " No Polisi: " << P->info.nopol << endl;
39         cout << " Warna : " << P->info.warna << endl;
40         cout << " Tahun : " << P->info.thnBuat << endl;
41         cout << "-----\\n";
42         P = P->next;
43         count++;
44     }
45 }
46
47 void insertLast(List &L, address P) {
48     if (L.First == nullptr) {
49         L.First = P;
50         L.Last = P;
51     } else {
52         L.Last->next = P;
53         P->prev = L.Last;
54         L.Last = P;
55     }
56 }
57
58 bool isNopolExist(List L, std::string nopol) {
59     address P = L.First;
60
61     while (P != nullptr) {
62         if (P->info.nopol == nopol) {
63             return true;
64         }
65         P = P->next;
66     }
67
68     return false; // Nopol tidak ditemukan

```

Kendaraan.h

```
Modul 6 > C kendaraan.h > kendaraan_H
 1  #ifndef kendaraan_H
 2  #define kendaraan_H
 3
 4  #include <string>
 5
 6  struct Kendaraan {
 7      std::string nopol;
 8      std::string warna;
 9      int thnBuat;
10  };
11
12  struct EImList {
13     Kendaraan info;
14     EImList* next;
15     EImList* prev;
16  };
17
18  typedef EImList* address;
19
20  struct List {
21     address First;
22     address Last;
23  };
24
25  void CreateList(List &L);
26  address alokasi(Kendaraan x);
27  void dealokasi(address &P);
28  void printInfo(List L);
29  void insertLast(List &L, address P);
30  bool isNopolExist(List L, std::string nopol);
31
32  #endif
```

Screenshots Output

```
==== MENU MANAJEMEN KENDARAAN ====
1. Tambah Data Kendaraan
2. Tampilkan Semua Data
3. Keluar
Pilihan: 1
```

```
==== MENU MANAJEMEN KENDARAAN ====
1. Tambah Data Kendaraan
2. Tampilkan Semua Data
3. Keluar
Pilihan: 1

--- Input Data Kendaraan ---
Masukkan nomor polisi: 0011
Masukkan warna kendaraan: Blue
Masukkan tahun kendaraan: 88

Data kendaraan berhasil ditambahkan!
```

==== MENU MANAJEMEN KENDARAAN ===

1. Tambah Data Kendaraan
 2. Tampilkan Semua Data
 3. Keluar
- Pilihan: 2

DATA LIST

Kendaraan #1:

No Polisi: 0011
Warna : Blue
Tahun : 88
