

**LAPORAN PRAKTIKUM  
MODUL 6  
DOUBLE LINKED LIST BAGIAN 1**



**Disusun Oleh:  
Yehuda Melvin Sugiarto  
SE-07-02**

**Dosen :  
Wahyu Andi Saputra**

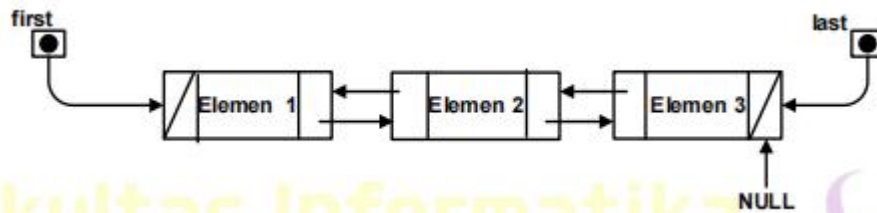
**PROGRAM STUDI S1 SOFTWARE ENGINEERING  
FAKULTAS INFORMATIKA  
TELKOM UNIVERSITY  
PURWOKERTO  
2024**

## Tujuan

1. Memahami konsep modul linked list.
2. Mengaplikasikan konsep double linked list dengan menggunakan pointer dan dengan bahasa C

## II. Landasan Teori

- Double Linked List merupakan list yang setiap elemennya memiliki 2 successor, prev (sebelum) dan next (sesudahnya).



Double Linked List memiliki beberapa komponen didalamnya, seperti:

First : pointer yang menunjuk elemen pertama dalam list

Last : pointer yang menunjuk elemen terakhir dalam list

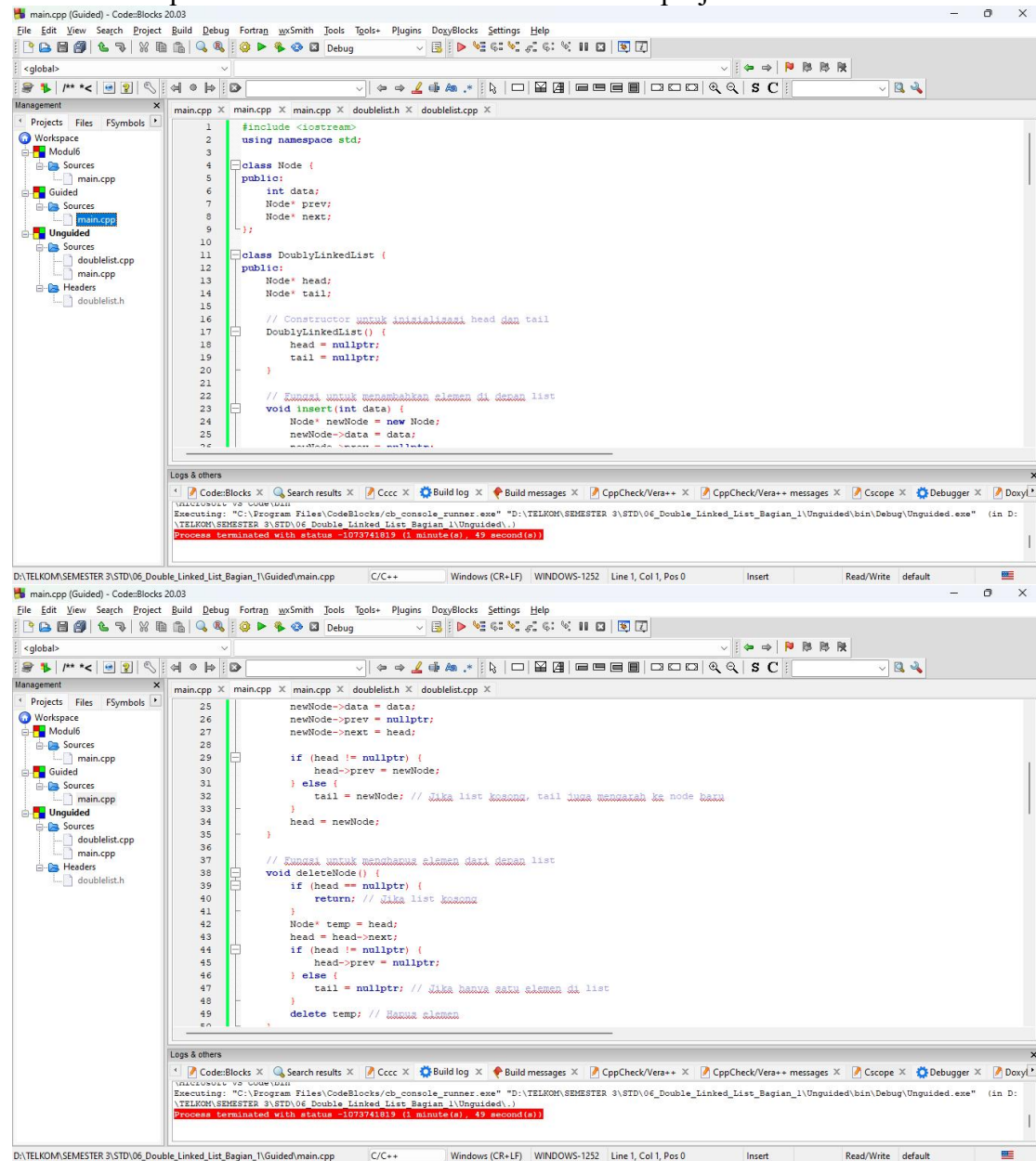
Next : pointer pada elemen sebagai successor yang menunjuk elemen setelahnya

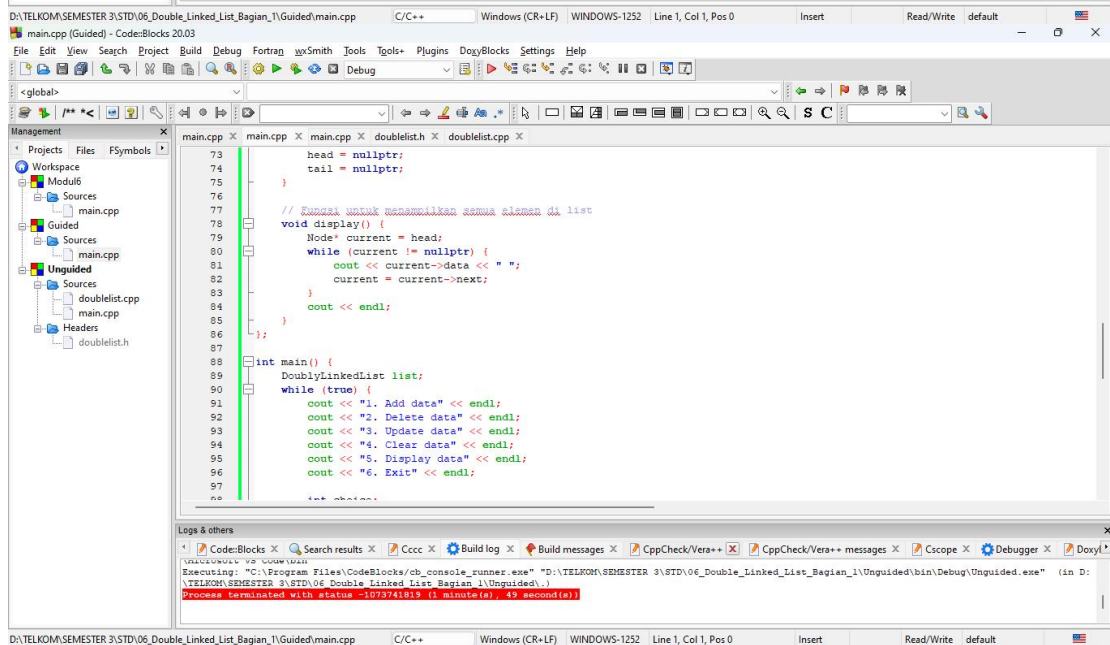
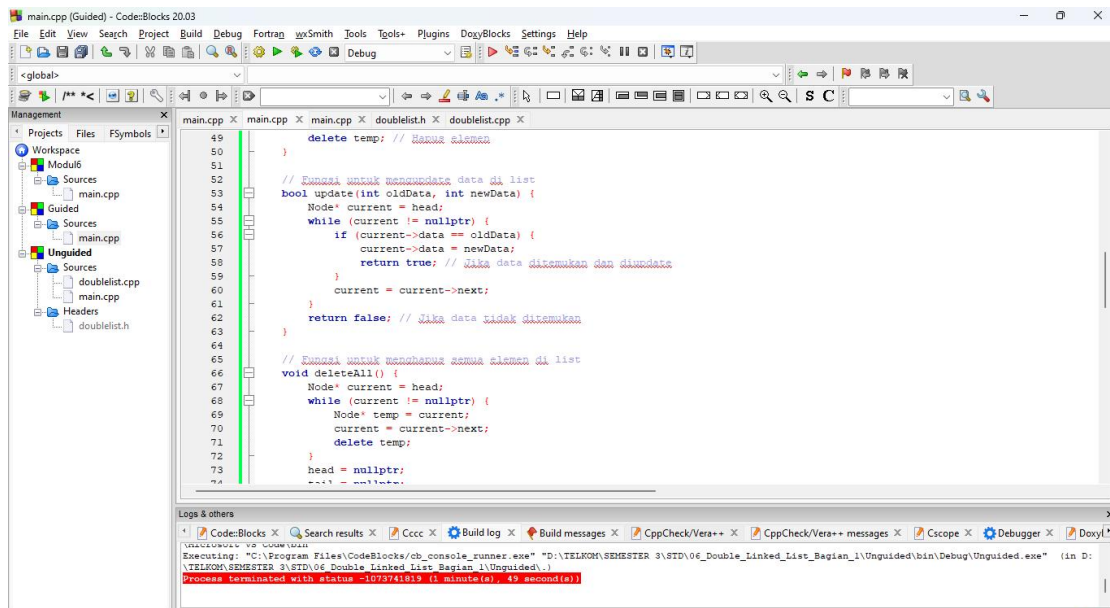
Prev : pointer pada elemen sebagai successor yang menunjuk elemen sebelumnya

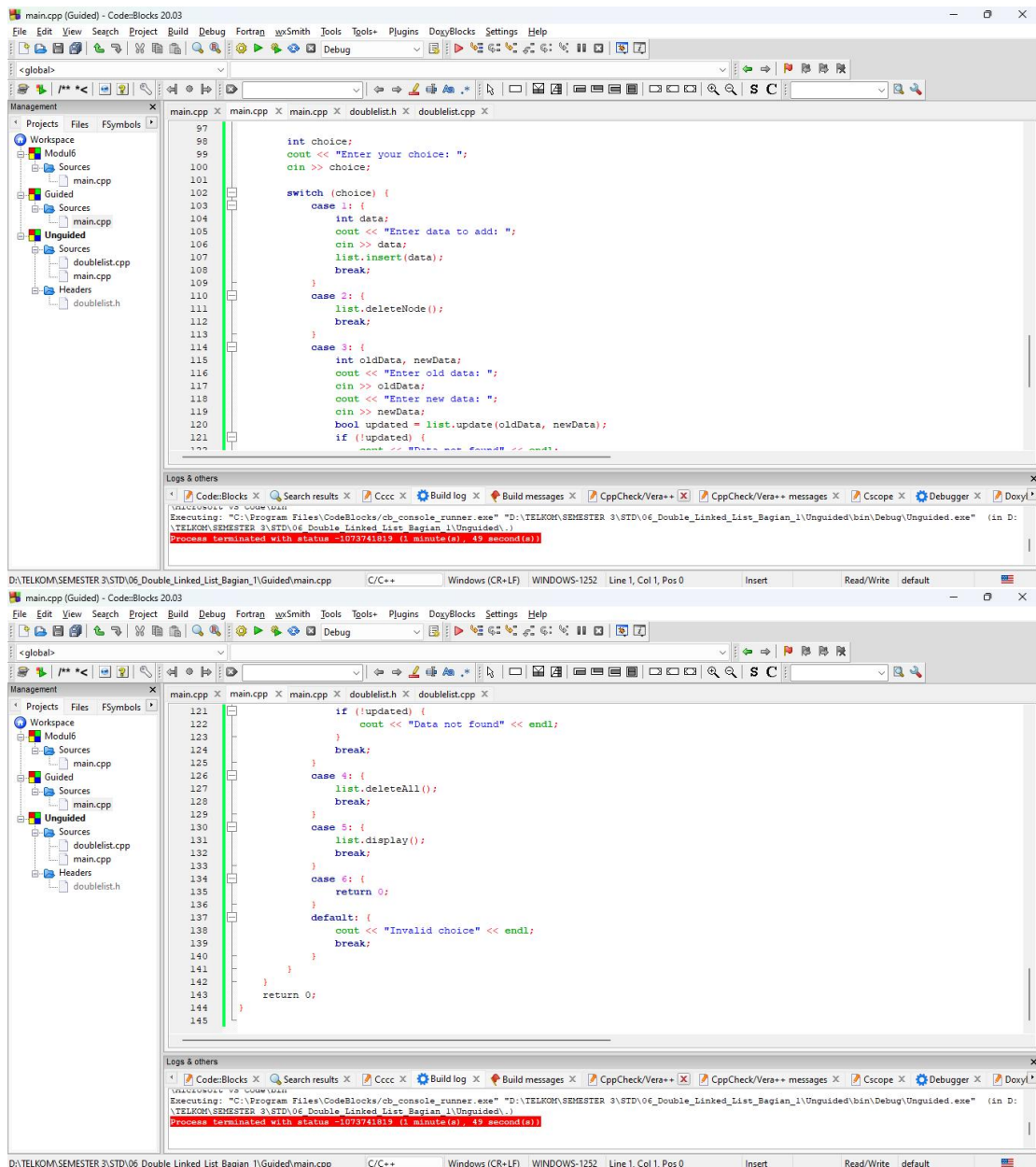
- Insert  
Merupakan method yang digunakan untuk menambahkan elemen pada list, terdapat beberapa method atau cara, seperti  
Insert First : menambah elemen pada awal list  
Insert Last : menambah elemen pada akhir list  
Insert After : menambah elemen setelah elemen tertentu  
Insert Before : menambah elemen sebelum elemen tertentu
- Delete  
Merupakan method yang digunakan untuk menghapus elemen pada list, sama seperti Insert, terdapat 4 cara yang dapat digunakan, yang membedakan hanyalah fungsinya.
- Update  
Merupakan method yang digunakan untuk mengubah informasi pada suatu elemen
- View  
Merupakan method yang digunakan untuk menampilkan informasi elemen didalam list
- Searching  
Merupakan method yang digunakan untuk mencari informasi elemen tertentu dalam suatu list

# III. Guided

## 1. Contoh Implementasi Method diatas kedalam suatu project







berikut suatu contoh implementasi dari method yang ada, pertama kita buat constructor untuk mendeklarasikan node, lalu constructor untuk deklarai double linked list. Setelah itu, kita buat beberapa method seperti Insert, Delete, Update, dan View yang dimana kita bisa memilih akan melakukan method apa saat kita run project

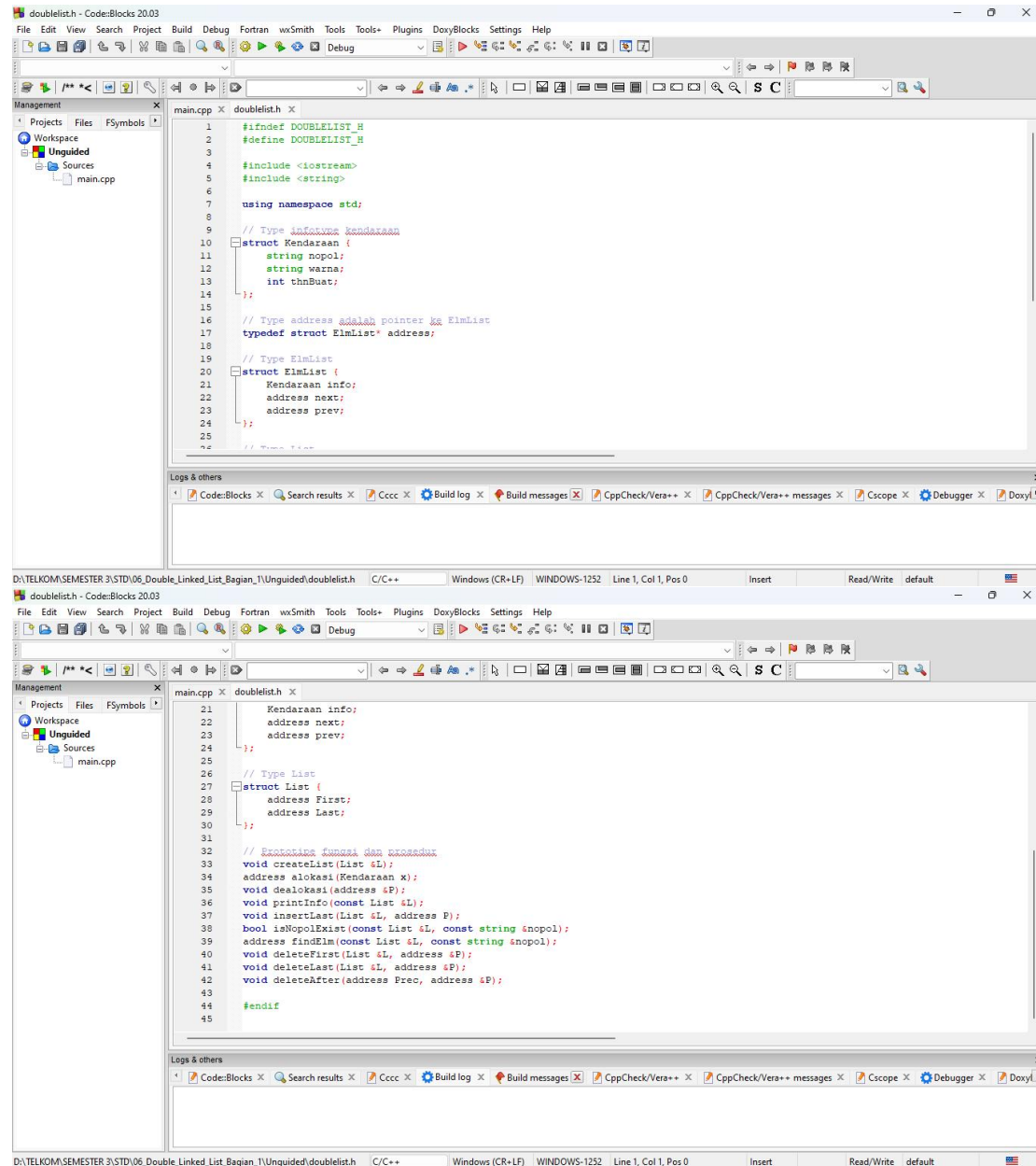
```
D:\TELKOM\SEMESTER 3\STC x + v
1. Add data
2. Delete data
3. Update data
4. Clear data
5. Display data
6. Exit
Enter your choice: 1
Enter data to add: 513531
1. Add data
2. Delete data
3. Update data
4. Clear data
5. Display data
6. Exit
Enter your choice: 1
Enter data to add: 53234243
1. Add data
2. Delete data
3. Update data
4. Clear data
5. Display data
6. Exit
Enter your choice: 3
Enter old data: 513531
Enter new data: 12
1. Add data
2. Delete data
3. Update data
4. Clear data
5. Display data
6. Exit
Enter your choice: 5
53234243 12
1. Add data
2. Delete data
3. Update data
4. Clear data

D:\TELKOM\SEMESTER 3\STC x + v
3. Update data
4. Clear data
5. Display data
6. Exit
Enter your choice: 1
Enter data to add:
532
1. Add data
2. Delete data
3. Update data
4. Clear data
5. Display data
6. Exit
Enter your choice: 2
1. Add data
2. Delete data
3. Update data
4. Clear data
5. Display data
6. Exit
Enter your choice: 5
53234243 12
1. Add data
2. Delete data
3. Update data
4. Clear data
5. Display data
6. Exit
Enter your choice: 4
1. Add data
2. Delete data
3. Update data
4. Clear data
5. Display data
6. Exit
Enter your choice: 5
```

## IV. UNGUIDED

1. Membuat project untuk membuat double linked list yang tiap elemen berisi informasi nopol, warna, dan tahun pembuatan. Dan kita juga akan menambahkan beberapa method seperti Insert, Delete, Searching, dan View.

File doublelist.H



```
1 #ifndef DOUBBLELIST_H
2 #define DOUBBLELIST_H
3
4 #include <iostream>
5 #include <string>
6
7 using namespace std;
8
9 // Type informasi kendaraan
10 struct Kendaraan {
11     string nopol;
12     string warna;
13     int thnBuat;
14 };
15
16 // Type address adalah pointer ke ElmList
17 typedef struct ElmList* address;
18
19 // Type ElmList
20 struct ElmList {
21     Kendaraan info;
22     address next;
23     address prev;
24 };
25
26 // Fungsi untuk membuat list
27 void createList(List &L);
28 void alokasi(Kendaraan x);
29 void dealokasi(address &P);
30 void printInfo(const List &L);
31 void insertLast(List &L, address P);
32 bool isNopolExist(const List &L, const string &nopol);
33 address findElm(const List &L, const string &nopol);
34 void deleteFirst(List &L, address &P);
35 void deleteLast(List &L, address &P);
36 void deleteAfter(address &P, address &P);
37
38 #endif
```



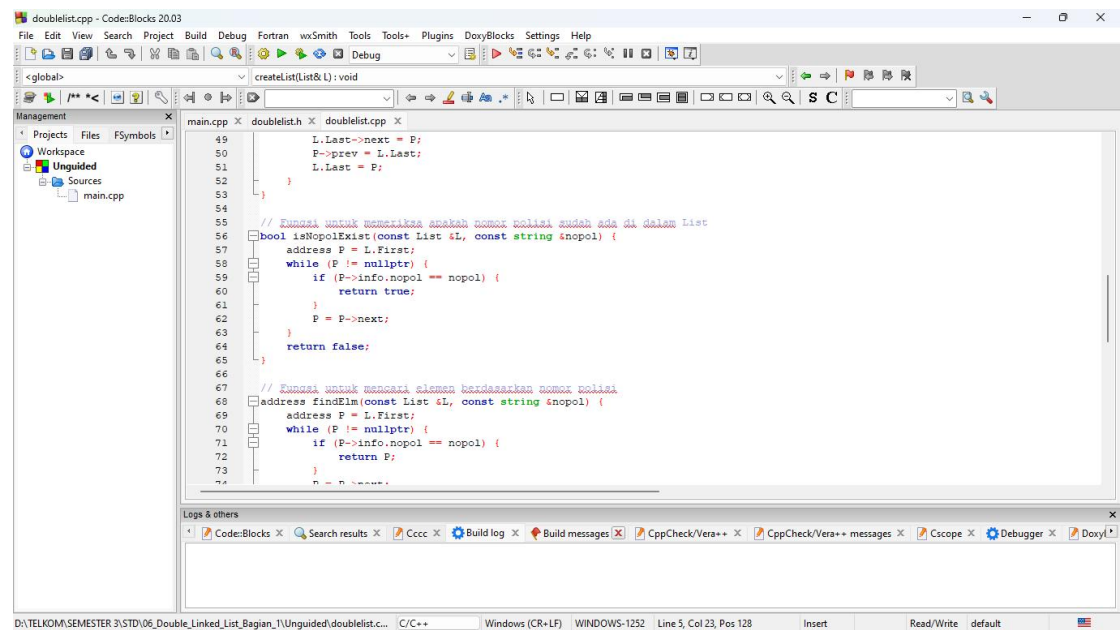
## File doublelist.cpp

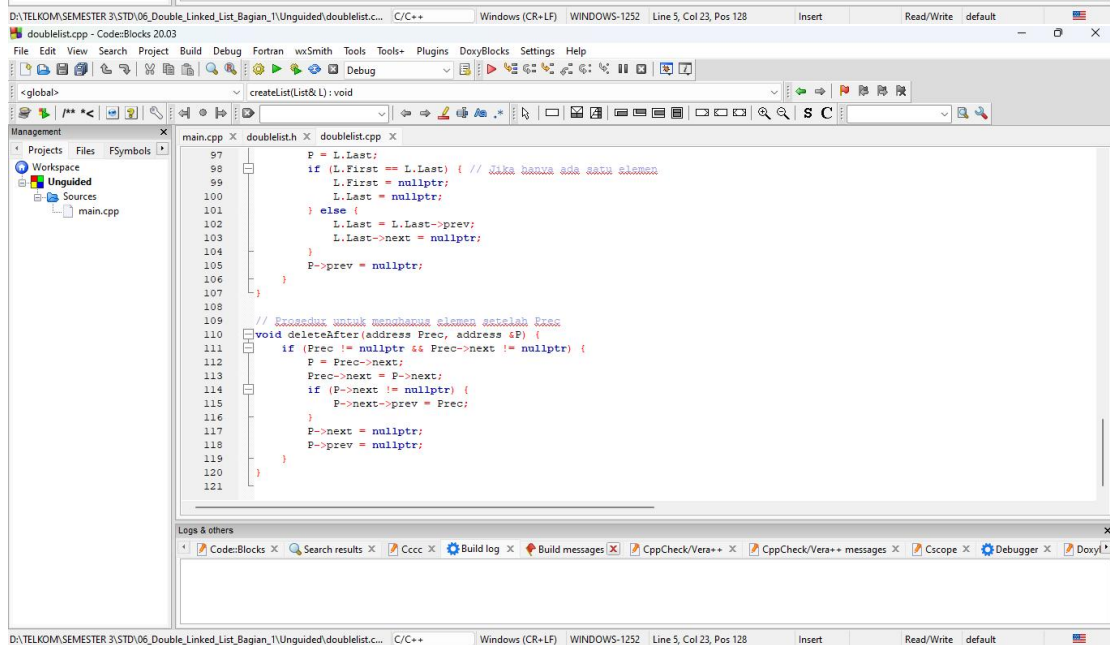
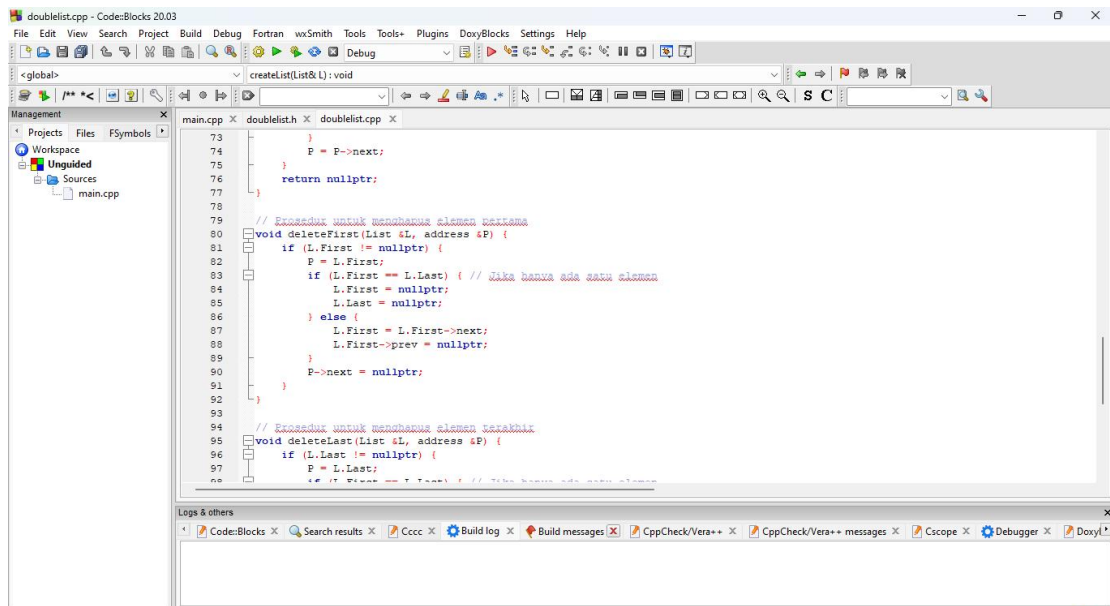
```
doublelist.cpp - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
<global> createList(List& L):void

main.cpp doublelist.h doublelist.cpp
1 #include "doublelist.h"
2
3 // Prosedur untuk membuat List baru (List kosong)
4 void createList(List& L) {
5     L.First = nullptr;
6     L.Last = nullptr;
7 }
8
9 // Prosedur untuk menambahkan elemen baru dengan data kendaraan
10 address alokasi(Kendaraan x) {
11     address P = new ElemList;
12     P->info = x;
13     P->next = nullptr;
14     P->prev = nullptr;
15     return P;
16 }
17
18 // Prosedur untuk menghapus elemen (menghapus node)
19 void dealokasi(address& P) {
20     delete P;
21     P = nullptr;
22 }
23
24 // Prosedur untuk menampilkan seluruh informasi dalam List
25 void printInfo(const List& L) {
26     address P = L.First;
27     if (P == nullptr) {
28         cout << "List kosong" << endl;
29     } else {
30         cout << "DATA LIST" << endl;
31         int i = 1;
32         while (P != nullptr) {
33             cout << i << ".\n";
34             cout << "No Polisi : " << P->info.nopol << endl;
35             cout << "Warna : " << P->info.warna << endl;
36             cout << "Tahun : " << P->info.thnBuat << endl;
37             P = P->next;
38             i++;
39         }
40     }
41 }
42
43 // Prosedur untuk memasukkan elemen baru di akhir List
44 void insertLast(List& L, address P) {
45     if (L.First == nullptr) {
46         L.First = P;
47         L.Last = P;
48     } else {
49         L.Last->next = P;
50         P->prev = L.Last;
51         L.Last = P;
52     }
53 }
```

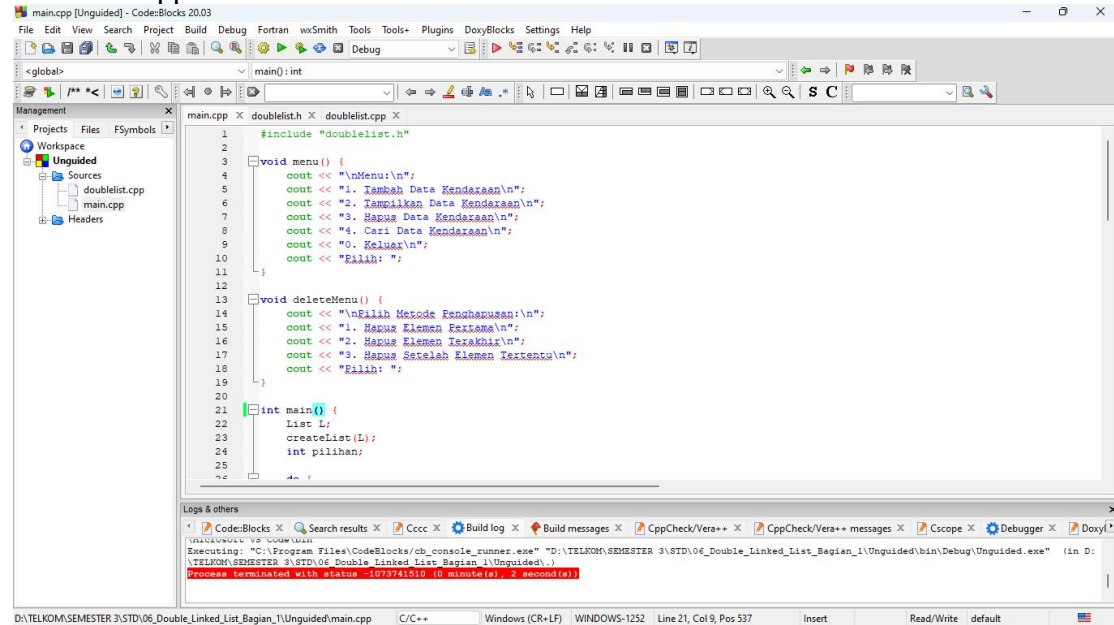
D:\TELKOM\SEMESTER 3\STD\06\_Double\_Linked\_List\_Bagian\_1\Unguided\doublelist.c... C/C++ Windows (CR+LF) WINDOWS-1252 Line 5, Col 23, Pos 128 Insert Read/Write default







## File main.cpp



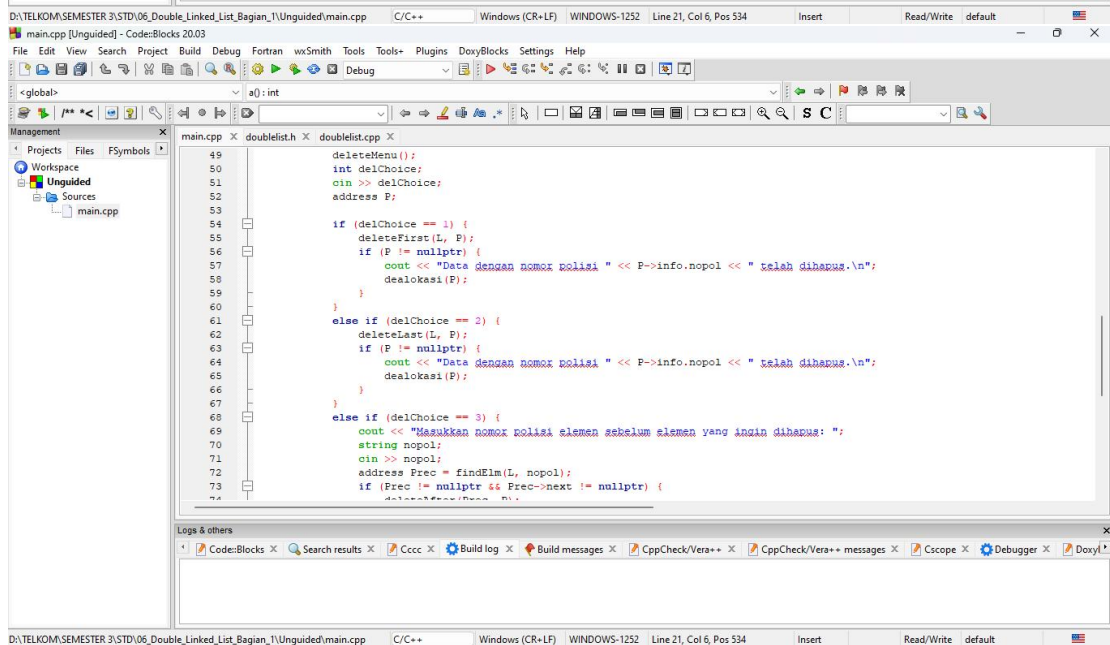
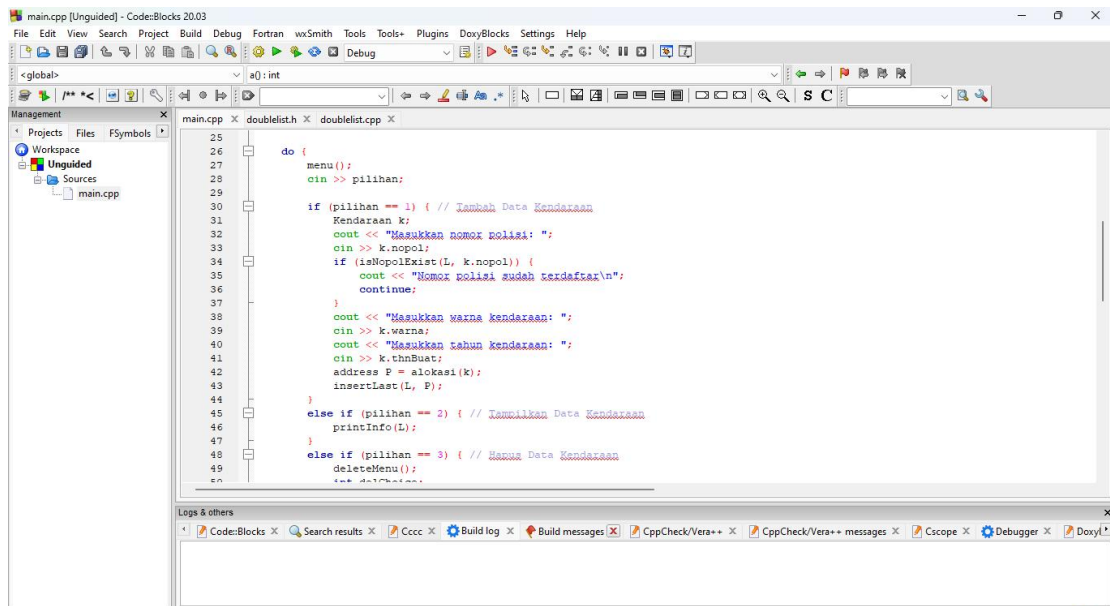
The screenshot displays the Code::Blocks IDE interface. The main editor window shows the `main.cpp` file with the following code:

```
1 #include "doublelist.h"
2
3
4 void menu() {
5     cout << "\nMenu:\n";
6     cout << "1. Tambah Data Kendaraan\n";
7     cout << "2. Tampilkan Data Kendaraan\n";
8     cout << "3. Hapus Data Kendaraan\n";
9     cout << "4. Cari Data Kendaraan\n";
10    cout << "0. Keluar\n";
11    cout << "Pilih: ";
12}
13
14 void deleteMenu() {
15     cout << "\nPilih Metode Penghapusan:\n";
16     cout << "1. Hapus Elemen Pertama\n";
17     cout << "2. Hapus Elemen Terakhir\n";
18     cout << "3. Hapus Setelah Elemen Tertentu\n";
19     cout << "Pilih: ";
20}
21
22 int main() {
23     List L;
24     createList(L);
25     int pilihan;
```

The left sidebar shows the project structure with `doublelist.cpp` and `main.cpp` under the `Sources` folder. The bottom status bar indicates the file path: `D:\TELKOM\SEMESTER 3\STD\06_Double_Linked_List_Bagian_1\Unguided\main.cpp`.

The `Logs & others` panel at the bottom shows the execution output:

```
Executing: "C:\Program Files\CodeBlocks\cb_console_runner.exe" "D:\TELKOM\SEMESTER 3\STD\06_Double_Linked_List_Bagian_1\Unguided\bin\Debug\Unguided.exe" (in D:\TELKOM\SEMESTER 3\STD\06_Double_Linked_List_Bagian_1\Unguided\bin\Debug)
Process terminated with status -1073741510 (0 minute(s), 2 second(s))
```



```

74         deleteAfter(Prec, P);
75         cout << "Data dengan nomor polisi " << P->info.nopol << " telah dihapus.\n";
76         dealokasi(P);
77     } else {
78         cout << "Elemen tidak ditemukan atau tidak ada elemen setelahnya.\n";
79     }
80 }
81
82 else if (pilihan == 4) { // Cari Data Kendaraan
83     cout << "Masukkan nomor polisi yang dicari: ";
84     string nopol;
85     cin >> nopol;
86     address P = findElem(L, nopol);
87     if (P != nullptr) {
88         cout << "Data ditemukan:\n";
89         cout << "No polisi : " << P->info.nopol << endl;
90         cout << "Warna : " << P->info.warna << endl;
91         cout << "Tahun : " << P->info.thnBuat << endl;
92     } else {
93         cout << "Data tidak ditemukan.\n";
94     }
95 }
96
97 } while (pilihan != 0);
98
99
100
101

```

## Output:

```

Menu:
1. Tambah Data Kendaraan
2. Tampilkan Data Kendaraan
3. Hapus Data Kendaraan
4. Cari Data Kendaraan
0. Keluar
Pilih: 1
Masukkan nomor polisi: D001
Masukkan warna kendaraan: hitam
Masukkan tahun kendaraan: 90

Menu:
1. Tambah Data Kendaraan
2. Tampilkan Data Kendaraan
3. Hapus Data Kendaraan
4. Cari Data Kendaraan
0. Keluar
Pilih: 1
Masukkan nomor polisi: D003
Masukkan warna kendaraan: putih
Masukkan tahun kendaraan: 70

Menu:
1. Tambah Data Kendaraan
2. Tampilkan Data Kendaraan
3. Hapus Data Kendaraan
4. Cari Data Kendaraan
0. Keluar
Pilih: 1
Masukkan nomor polisi: D004
Masukkan warna kendaraan: kuning
Masukkan tahun kendaraan: 90

```

```
D:\TELKOM\SEMESTER 3\STC x + v
1. Tambah Data Kendaraan
2. Tampilkan Data Kendaraan
3. Hapus Data Kendaraan
4. Cari Data Kendaraan
0. Keluar
Pilih: 2
DATA LIST
1.
No polisi : D001
Warna : hitam
Tahun : 90
2.
No polisi : D003
Warna : putih
Tahun : 70
3.
No polisi : D004
Warna : kuning
Tahun : 90

Menu:
1. Tambah Data Kendaraan
2. Tampilkan Data Kendaraan
3. Hapus Data Kendaraan
4. Cari Data Kendaraan
0. Keluar
Pilih: 1
Masukkan nomor polisi: D001
Nomor polisi sudah terdaftar

Menu:
1. Tambah Data Kendaraan
2. Tampilkan Data Kendaraan
3. Hapus Data Kendaraan
4. Cari Data Kendaraan
0. Keluar
Pilih: |
```

```
D:\TELKOM\SEMESTER 3\STC x + v
3.
No polisi : D004
Warna : kuning
Tahun : 90

Menu:
1. Tambah Data Kendaraan
2. Tampilkan Data Kendaraan
3. Hapus Data Kendaraan
4. Cari Data Kendaraan
0. Keluar
Pilih: 1
Masukkan nomor polisi: D001
Nomor polisi sudah terdaftar

Menu:
1. Tambah Data Kendaraan
2. Tampilkan Data Kendaraan
3. Hapus Data Kendaraan
4. Cari Data Kendaraan
0. Keluar
Pilih: 3

Pilih Metode Penghapusan:
1. Hapus Elemen Pertama
2. Hapus Elemen Terakhir
3. Hapus Setelah Elemen Tertentu
Pilih: 1
Data dengan nomor polisi D001 telah dihapus.

Menu:
1. Tambah Data Kendaraan
2. Tampilkan Data Kendaraan
3. Hapus Data Kendaraan
4. Cari Data Kendaraan
0. Keluar
Pilih: |

D:\TELKOM\SEMESTER 3\STC x + v
Pilih Metode Penghapusan:
1. Hapus Elemen Pertama
2. Hapus Elemen Terakhir
3. Hapus Setelah Elemen Tertentu
Pilih: 1
Data dengan nomor polisi D001 telah dihapus.

Menu:
1. Tambah Data Kendaraan
2. Tampilkan Data Kendaraan
3. Hapus Data Kendaraan
4. Cari Data Kendaraan
0. Keluar
Pilih: 4
Masukkan nomor polisi yang dicari: D003
Data ditemukan:
No polisi : D003
Warna : putih
Tahun : 70

Menu:
1. Tambah Data Kendaraan
2. Tampilkan Data Kendaraan
3. Hapus Data Kendaraan
4. Cari Data Kendaraan
0. Keluar
Pilih: 4
Masukkan nomor polisi yang dicari: D001
Data tidak ditemukan.

Menu:
1. Tambah Data Kendaraan
2. Tampilkan Data Kendaraan
3. Hapus Data Kendaraan
4. Cari Data Kendaraan
0. Keluar
Pilih: |
```

## VI. Kesimpulan

Praktikum kali ini dapat disimpulkan bahwa double linked list merupakan list dengan 2 successor (prev dan next), kita juga mengimplementasikan method method yang ada pada linked list dan jenis jenis insert dan delete.