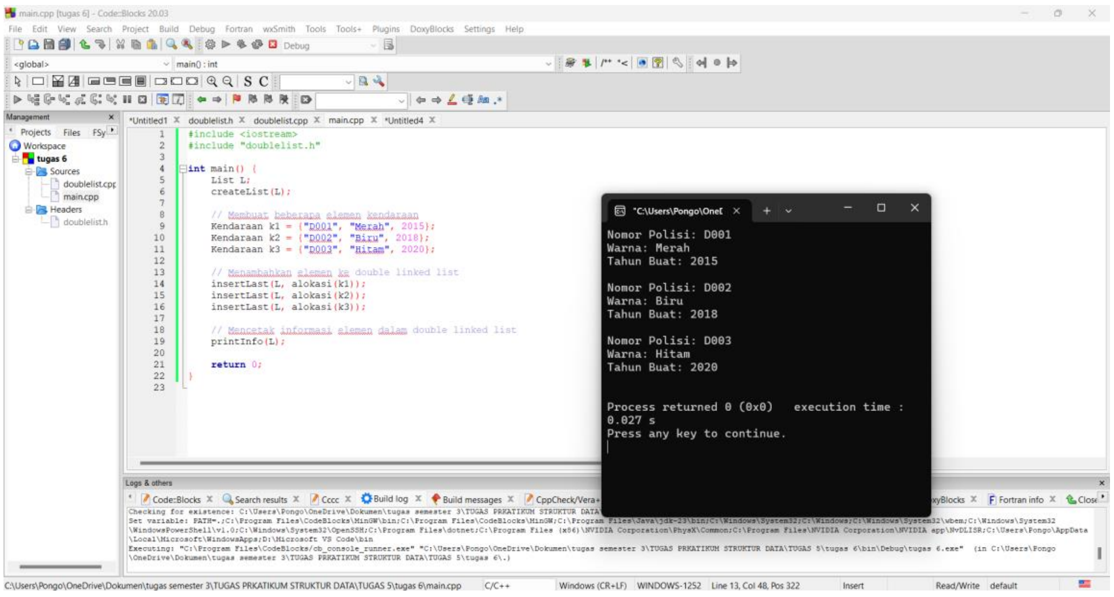
UNGUIDED

1. doubblist.h

```
#ifndef DOUBLELIST_H
#define DOUBLELIST_H

#include <string>

struct Kendaraan {
	std::string nopol;
	std::string warna;
	int thnBuat;
};
typedef Kendaraan infotype;

typedef struct ElmList *address;

struct ElmList {
	infotype info;
	address next;
	address prev;
};

struct List {
	address first;
	address last;
};

void createList(List& L);
address alokasi(infotype x);
void dealokasi(address& P);
void printInfo(List L);
void insertLast(List& L, address P);

#endif // DOUBLELIST_H
```

main.cpp

```
#include <iostream>
#include "doublelist.h"

int main() {
	List L;
	createList(L);

	// Membuat beberapa elemen kendaraan
	Kendaraan k1 = {"D001", "Merah", 2015};
	Kendaraan k2 = {"D002", "Biru", 2018};
	Kendaraan k3 = {"D003", "Hitam", 2020};

	// Menambahkan elemen ke double linked list
	insertLast(L, alokasi(k1));
	insertLast(L, alokasi(k2));
	insertLast(L, alokasi(k3));

	// Mencetak informasi elemen dalam double linked list
	printInfo(L);

	return 0;
}

doublelist.cpp
#include "doublelist.h"
#include <iostream>

void createList(List& L) {
	L.first = nullptr;
	L.last = nullptr;
}

address alokasi(infotype x) {
	address P = new ElmList;
	P->info = x;
	P->next = nullptr;
	P->prev = nullptr;
	return P;
}

void dealokasi(address& P) {
	delete P;
	P = nullptr;
}

void printInfo(List L) {
	address P = L.first;
	while (P != nullptr) {
		std::cout << "Nomor Polisi: " << P->info.nopol << std::endl;
		std::cout << "Warna: " << P->info.warna << std::endl;
		std::cout << "Tahun Buat: " << P->info.thnBuat << std::endl;
		std::cout << std::endl;
		P = P->next;
	}
}

void insertLast(List& L, address P) {
	if (L.first == nullptr) {
		L.first = P;
		L.last = P;
	} else {
		L.last->next = P;
		P->prev = L.last;
		L.last = P;
	}
}
```



```cpp
#include <iostream>
#include "doublelist.h"

int main() {
    List L;
    createList(L);

    // Membuat beberapa elemen kendaraan
    Kendaraan k1 = {"D001", "Merah", 2015};
    Kendaraan k2 = {"D002", "Biru", 2018};
    Kendaraan k3 = {"D003", "Hitam", 2020};

    // Menambahkan elemen ke double linked list
    insertLast(L, alokasi(k1));
    insertLast(L, alokasi(k2));
    insertLast(L, alokasi(k3));

    // Mencetak informasi elemen dalam double linked list
    printInfo(L);

    return 0;
}
```

```
Nomor Polisi: D001
Warna: Merah
Tahun Buat: 2015

Nomor Polisi: D002
Warna: Biru
Tahun Buat: 2018

Nomor Polisi: D003
Warna: Hitam
Tahun Buat: 2020


Process returned 0 (0x0)   execution time :
0.027 s
Press any key to continue.
```

```
Checking for existence: C:\Users\Pongo\OneDrive\Dokumen\tugas semester 3\TUGAS PRKATIKUM STRUKTUR DATA
Set variable: PATH=.;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\CodeBlocks\MinGW;C:\Program Files\Java\jdk-23\bin;C:\Windows\System32;C:\Windows;C:\Windows\System32\wbem;C:\Windows\System32
\WindowsPowerShell\v1.0;C:\Windows\System32\OpenSSH;C:\Program Files\dotnet;C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common;C:\Program Files\NVIDIA Corporation\NVIDIA app\NvDLISR;C:\Users\Pongo\AppData
\Local\Microsoft\WindowsApps;D:\Microsoft VS Code\bin
Executing: "C:\Program Files\CodeBlocks/cb_console_runner.exe" "C:\Users\Pongo\OneDrive\Dokumen\tugas semester 3\TUGAS PRKATIKUM STRUKTUR DATA\TUGAS 5\tugas 6\bin\Debug\tugas 6.exe"  (in C:\Users\Pongo
\OneDrive\Dokumen\tugas semester 3\TUGAS PRKATIKUM STRUKTUR DATA\TUGAS 5\tugas 6\.)
```

2. Doublelist.h

```cpp
#ifndef DOUBLELIST_H
#define DOUBLELIST_H

#include <string>

struct Kendaraan {
        std::string nopol;
        std::string warna;
        int thnBuat;
};
typedef Kendaraan infotype;

typedef struct ElmList *address;

struct ElmList {
        infotype info;
        address next;
        address prev;
};

struct List {
        address first;
        address last;
};

void createList(List& L);
address alokasi(infotype x);
void dealokasi(address& P);
void printInfo(List L);
void insertLast(List& L, address P);
address findElm(List L, infotype x);

#endif // DOUBLELIST_H
```

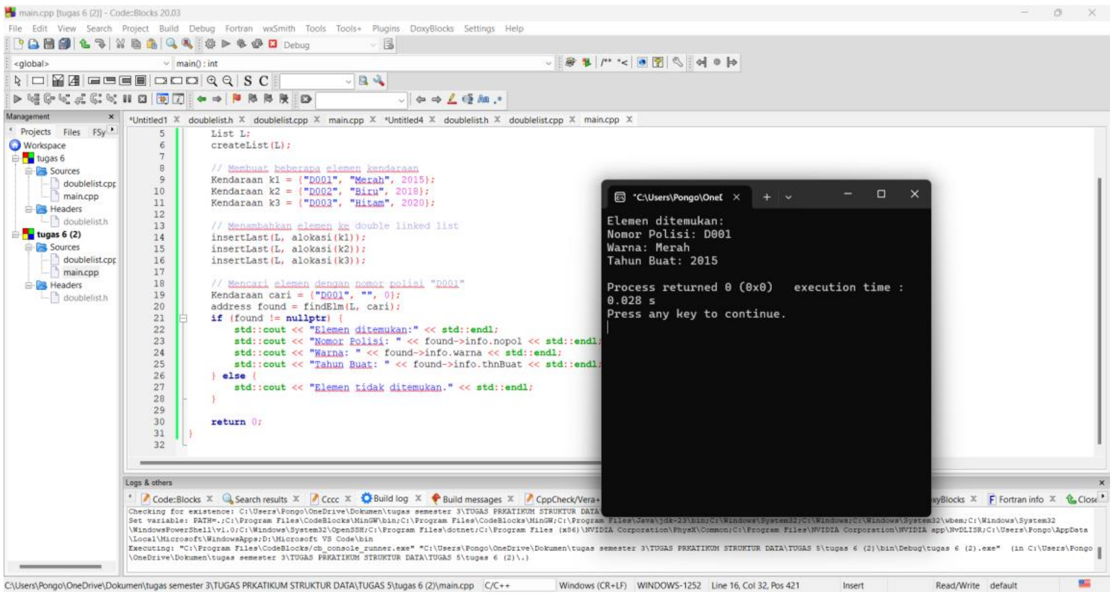
Doublelist.cpp

```cpp
#include "doublelist.h"
#include <iostream>
```

```
void createList(List& L) {
        L.first = nullptr;
        L.last = nullptr;
}

address alokasi(infotype x) {
        address P = new ElmList;
        P->info = x;
        P->next = nullptr;
        P->prev = nullptr;
        return P;
}

void dealokasi(address& P) {
        delete P;
        P = nullptr;
}

void printInfo(List L) {
        address P = L.first;
        while (P != nullptr) {
                std::cout << "Nomor Polisi: " << P->info.nopol << std::endl;
                std::cout << "Warna: " << P->info.warna << std::endl;
                std::cout << "Tahun Buat: " << P->info.thnBuat << std::endl;
                std::cout << std::endl;
                P = P->next;
        }
}

void insertLast(List& L, address P) {
        if (L.first == nullptr) {
                L.first = P;
                L.last = P;
        } else {
                L.last->next = P;
                P->prev = L.last;
                L.last = P;
        }
}

address findElm(List L, infotype x) {
        address P = L.first;
        while (P != nullptr) {
                if (P->info.nopol == x.nopol) {
                        return P;
                }
                P = P->next;
        }
        return nullptr;
}
```

Main.cpp

```
#include <iostream>
#include "doublelist.h"

int main() {
    List L;
    createList(L);

    // Membuat beberapa elemen kendaraan
    Kendaraan k1 = {"D001", "Merah", 2015};
    Kendaraan k2 = {"D002", "Biru", 2018};
    Kendaraan k3 = {"D003", "Hitam", 2020};

    // Menambahkan elemen ke double linked list
    insertLast(L, alokasi(k1));
    insertLast(L, alokasi(k2));
    insertLast(L, alokasi(k3));

    // Mencari elemen dengan nomor polisi "D001"
    Kendaraan cari = {"D001", "", 0};
    address found = findElm(L, cari);
    if (found != nullptr) {
        std::cout << "Elemen ditemukan:" << std::endl;
        std::cout << "Nomor Polisi: " << found->info.nopol << std::endl;
        std::cout << "Warna: " << found->info.warna << std::endl;
        std::cout << "Tahun Buat: " << found->info.thnBuat << std::endl;
    } else {
        std::cout << "Elemen tidak ditemukan." << std::endl;
    }

    return 0;
}
```


main.cpp [tugas 6 (2)] - Code::Blocks 20.03
File Edit View Search Project Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
<global>
main() : int
Management
Projects Files FSy
Workspace
tugas 6
Sources
doublelist.cpp
main.cpp
Headers
doublelist.h
tugas 6 (2)
Sources
doublelist.cpp
main.cpp
Headers
doublelist.h
*Untitled1 doublelist.h doublelist.cpp main.cpp *Untitled4 doublelist.h doublelist.cpp main.cpp
"C:\Users\Pongo\OneI
Logs & others
Code::Blocks Search results Cccc Build log Build messages CppCheck/Vera++ DoxyBlocks Fortran info Close
C/C++ Windows (CR+LF) WINDOWS-1252 Line 16, Col 32, Pos 421 Insert Read/Write default

```
    List L;
    createList(L);

    // Membuat beberapa elemen kendaraan
    Kendaraan k1 = {"D001", "Merah", 2015};
    Kendaraan k2 = {"D002", "Biru", 2018};
    Kendaraan k3 = {"D003", "Hitam", 2020};

    // Menambahkan elemen ke double linked list
    insertLast(L, alokasi(k1));
    insertLast(L, alokasi(k2));
    insertLast(L, alokasi(k3));

    // Mencari elemen dengan nomor polisi "D001"
    Kendaraan cari = {"D001", "", 0};
    address found = findElm(L, cari);
    if (found != nullptr) {
        std::cout << "Elemen ditemukan:" << std::endl;
        std::cout << "Nomor Polisi: " << found->info.nopol << std::endl
        std::cout << "Warna: " << found->info.warna << std::endl;
        std::cout << "Tahun Buat: " << found->info.thnBuat << std::endl
    } else {
        std::cout << "Elemen tidak ditemukan." << std::endl;
    }

    return 0;
}
```

```
Elemen ditemukan:
Nomor Polisi: D001
Warna: Merah
Tahun Buat: 2015

Process returned 0 (0x0)   execution time :
0.028 s
Press any key to continue.
```

```
Checking for existence: C:\Users\Pongo\OneDrive\Dokumen\tugas semester 3\TUGAS PRKATIKUM STRUKTUR DATA
Set variable: PATH=.;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\CodeBlocks\MinGW;C:\Program Files\Java\jdk-23\bin;C:\Windows\System32;C:\Windows;C:\Windows\System32\wbem;C:\Windows\System32
\WindowsPowerShell\v1.0;C:\Windows\System32\OpenSSH;C:\Program Files\dotnet;C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common;C:\Program Files\NVIDIA Corporation\NVIDIA app\NvDLISR;C:\Users\Pongo\AppData
\Local\Microsoft\WindowsApps;D:\Microsoft VS Code\bin
Executing: "C:\Program Files\CodeBlocks/cb_console_runner.exe" "C:\Users\Pongo\OneDrive\Dokumen\tugas semester 3\TUGAS PRKATIKUM STRUKTUR DATA\TUGAS 5\tugas 6 (2)\bin\Debug\tugas 6 (2).exe"  (in C:\Users\Pongo
\OneDrive\Dokumen\tugas semester 3\TUGAS PRKATIKUM STRUKTUR DATA\TUGAS 5\tugas 6 (2)\.)
```

C:\Users\Pongo\OneDrive\Dokumen\tugas semester 3\TUGAS PRKATIKUM STRUKTUR DATA\TUGAS 5\tugas 6 (2)\main.cpp

3. Doublelist.h

```
#ifndef DOUBLELIST_H
#define DOUBLELIST_H

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

struct infotype {
    string nopol;
```

```
        string warna;
        int thnBuat;
};

typedef struct ElmList* address;

struct ElmList {
        infotype info;
        address next;
        address prev;
};

struct List {
        address first;
        address last;
};

void CreateList(List& L);
address alokasi(infotype x);
void dealokasi(address& P);
void printInfo(List L);
void insertLast(List& L, address P);
address findElm(List L, infotype x);
void deleteAfter(List& L, address Prec, address& P);

#endif // DOUBLELIST_H
```

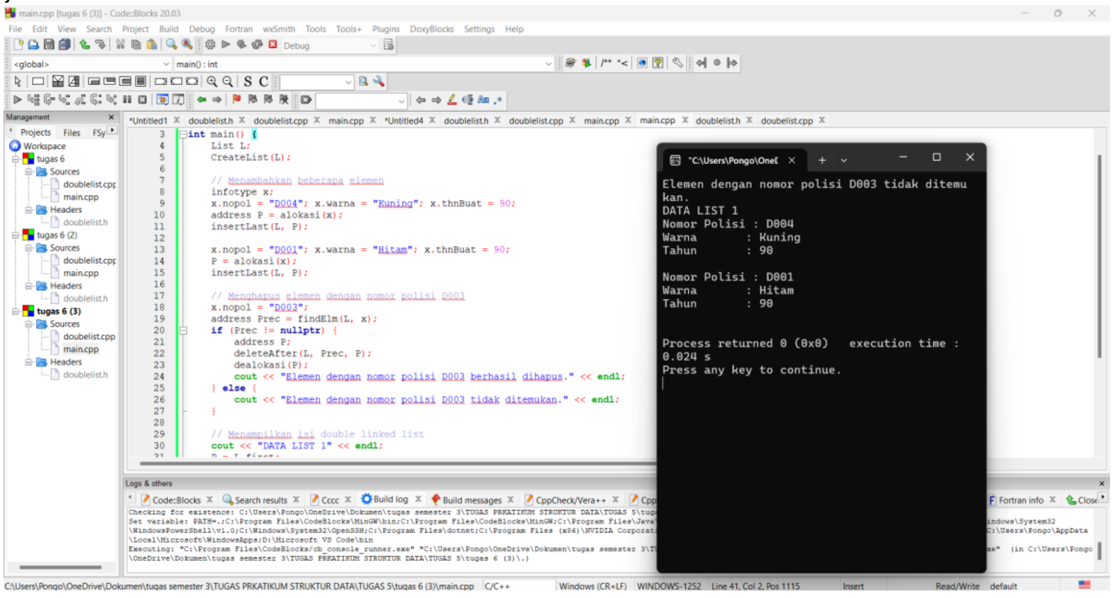
Doublelist.cpp

```
#include "doublelist.h"

void CreateList(List& L) {
        L.first = nullptr;
        L.last = nullptr;
}

address alokasi(infotype x) {
        address P = new ElmList;
        P->info = x;
        P->next = nullptr;
        P->prev = nullptr;
        return P;
}

void dealokasi(address& P) {
        delete P;
        P = nullptr;
}

void printInfo(List L) {
        address P = L.first;
        while (P != nullptr) {
                cout << "Nomor Polisi : " << P->info.nopol << endl;
                cout << "Warna        : " << P->info.warna << endl;
                cout << "Tahun        : " << P->info.thnBuat << endl;
                cout << endl;
                P = P->next;
        }
}

void insertLast(List& L, address P) {
	if (L.first == nullptr) {
		L.first = P;
		L.last = P;
	} else {
		L.last->next = P;
		P->prev = L.last;
		L.last = P;
	}
}

address findElm(List L, infotype x) {
	address P = L.first;
	while (P != nullptr && P->info.nopol != x.nopol) {
		P = P->next;
	}
	return P;
}

void deleteAfter(List& L, address Prec, address& P) {
	if (Prec != nullptr && Prec->next != nullptr) {
		P = Prec->next;
		Prec->next = P->next;
		if (P->next != nullptr) {
			P->next->prev = Prec;
		} else {
			L.last = Prec;
		}
		P->next = nullptr;
		P->prev = nullptr;
	} else {
		P = nullptr;
	}
}
```

Main.cpp

```
#include "doublelist.h"

int main() {
	List L;
	CreateList(L);

	// Menambahkan beberapa elemen
	infotype x;
	x.nopol = "D004"; x.warna = "Kuning"; x.thnBuat = 90;
	address P = alokasi(x);
	insertLast(L, P);

	x.nopol = "D001"; x.warna = "Hitam"; x.thnBuat = 90;
	P = alokasi(x);
	insertLast(L, P);

	// Menghapus elemen dengan nomor polisi D003
	x.nopol = "D003";
	address Prec = findElm(L, x);
```

```
        if (Prec != nullptr) {
                address P;
                deleteAfter(L, Prec, P);
                dealokasi(P);
                cout << "Elemen dengan nomor polisi D003 berhasil dihapus." << endl;
        } else {
                cout << "Elemen dengan nomor polisi D003 tidak ditemukan." << endl;
        }

        // Menampilkan isi double linked list
        cout << "DATA LIST 1" << endl;
        P = L.first;
        while (P != nullptr) {
                cout << "Nomor Polisi : " << P->info.nopol << endl;
                cout << "Warna              : " << P->info.warna << endl;
                cout << "Tahun              : " << P->info.thnBuat << endl;
                cout << endl;
                P = P->next;
        }

        return 0;
}
```



```
int main() {
    List L;
    CreateList(L);

    // Menambahkan beberapa elemen
    infotype x;
    x.nopol = "D004"; x.warna = "Kuning"; x.thnBuat = 90;
    address P = alokasi(x);
    insertLast(L, P);

    x.nopol = "D001"; x.warna = "Hitam"; x.thnBuat = 90;
    P = alokasi(x);
    insertLast(L, P);

    // Menghapus elemen dengan nomor polisi D003
    x.nopol = "D003";
    address Prec = findElm(L, x);
    if (Prec != nullptr) {
        address P;
        deleteAfter(L, Prec, P);
        dealokasi(P);
        cout << "Elemen dengan nomor polisi D003 berhasil dihapus." << endl;
    } else {
        cout << "Elemen dengan nomor polisi D003 tidak ditemukan." << endl;
    }

    // Menampilkan isi double linked list
    cout << "DATA LIST 1" << endl;
```

```
Elemen dengan nomor polisi D003 tidak ditemu
kan.
DATA LIST 1
Nomor Polisi : D004
Warna        : Kuning
Tahun        : 90

Nomor Polisi : D001
Warna        : Hitam
Tahun        : 90


Process returned 0 (0x0)   execution time :
0.024 s
Press any key to continue.
```

```
Checking for existence: C:\Users\Pongo\OneDrive\Dokumen\tugas semester 3\TUGAS PRKATIKUM STRUKTUR DATA\TUGAS 5\tug
Set variable: PATH=.;C:\Program Files\CodeBlocks\MinGW\bin;C:\Program Files\CodeBlocks\MinGW;C:\Program Files\Java
\WindowsPowerShell\v1.0;C:\Windows\System32\OpenSSH;C:\Program Files\dotnet;C:\Program Files (x86)\NVIDIA Corporat
\Local\Microsoft\WindowsApps;D:\Microsoft VS Code\bin
Executing: "C:\Program Files\CodeBlocks/cb_console_runner.exe" "C:\Users\Pongo\OneDrive\Dokumen\tugas semester 3\T
\OneDrive\Dokumen\tugas semester 3\TUGAS PRKATIKUM STRUKTUR DATA\TUGAS 5\tugas 6 (3)\.)
```

C:\Users\Pongo\OneDrive\Dokumen\tugas semester 3\TUGAS PRKATIKUM STRUKTUR DATA\TUGAS 5\tugas 6 (3)\main.cpp | C/C++ | Windows (CR+LF) | WINDOWS-1252 | Line 41, Col 2, Pos 1115 | Insert | Read/Write | default