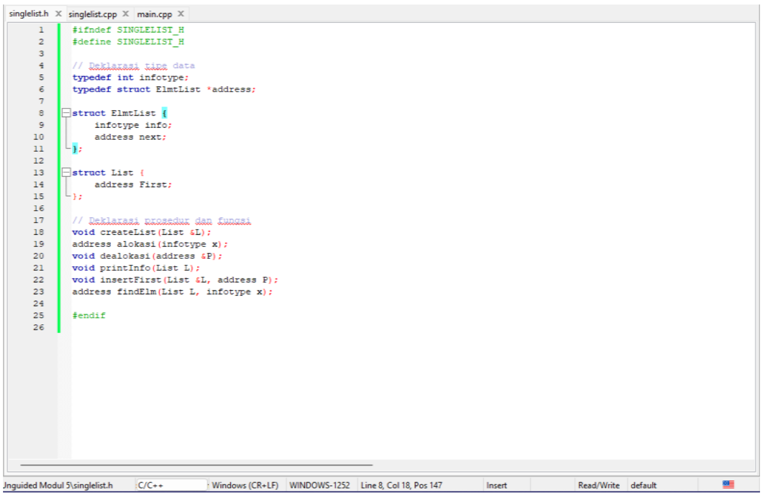
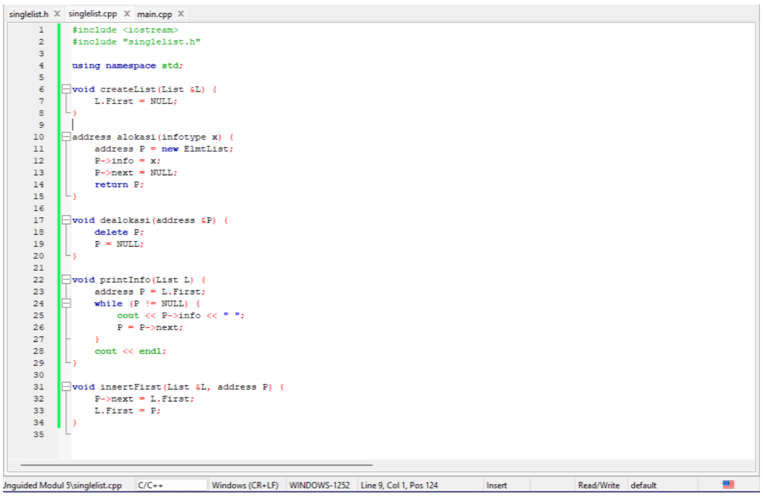
UNGUIDED LAPORAN PRAKTIKUM STUKTUR DATA MODUL 5

1. Unguided

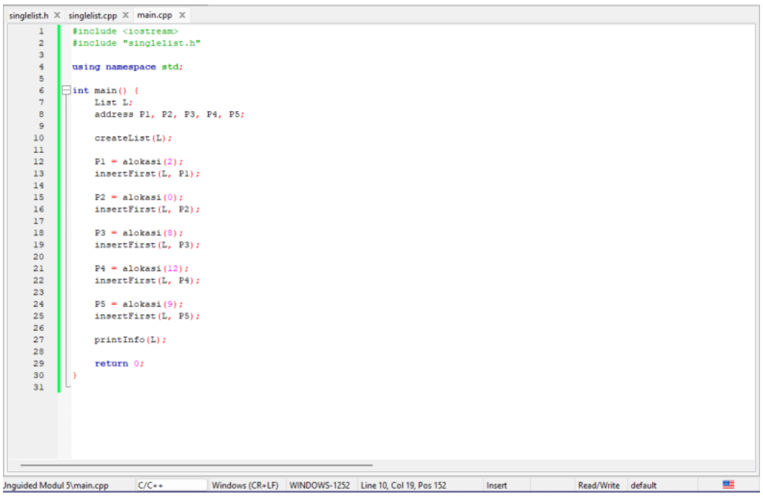
4.1. Kode Program :



```cpp
#ifndef SINGLELIST_H
#define SINGLELIST_H

// Deklarasi tipe data
typedef int infotype;
typedef struct ElmtList *address;

struct ElmtList {
    infotype info;
    address next;
};

struct List {
    address First;
};

// Deklarasi prosedur dan fungsi
void createList(List &L);
address alokasi(infotype x);
void dealokasi(address &P);
void printInfo(List L);
void insertFirst(List &L, address P);
address findElm(List L, infotype x);

#endif
```

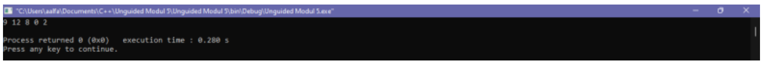


```cpp
#include <iostream>
#include "singlelist.h"

using namespace std;

void createList(List &L) {
    L.First = NULL;
}

address alokasi(infotype x) {
    address P = new ElmtList;
    P->info = x;
    P->next = NULL;
    return P;
}

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

void printInfo(List L) {
    address P = L.First;
    while (P != NULL) {
        cout << P->info << " ";
        P = P->next;
    }
    cout << endl;
}

void insertFirst(List &L, address P) {
    P->next = L.First;
    L.First = P;
}
```



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

using namespace std;

int main() {
    List L;
    address P1, P2, P3, P4, P5;

    createList(L);

    P1 = alokasi(2);
    insertFirst(L, P1);

    P2 = alokasi(0);
    insertFirst(L, P2);

    P3 = alokasi(8);
    insertFirst(L, P3);

    P4 = alokasi(12);
    insertFirst(L, P4);

    P5 = alokasi(9);
    insertFirst(L, P5);

    printInfo(L);

    return 0;
}
```

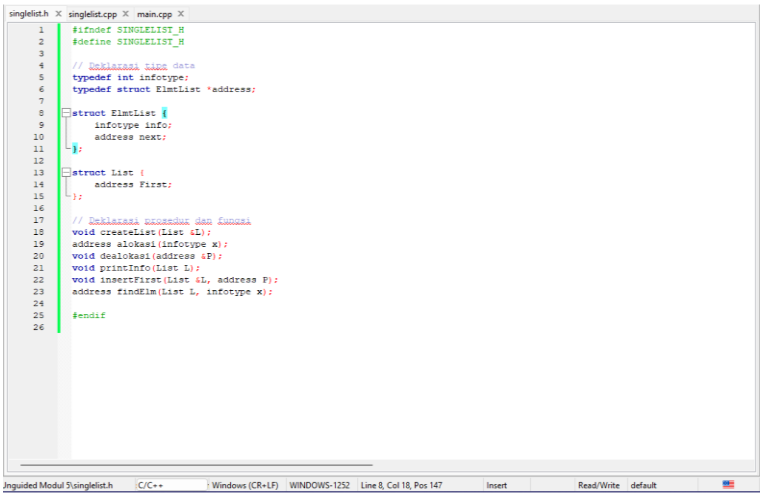
Output dari Kode Program :



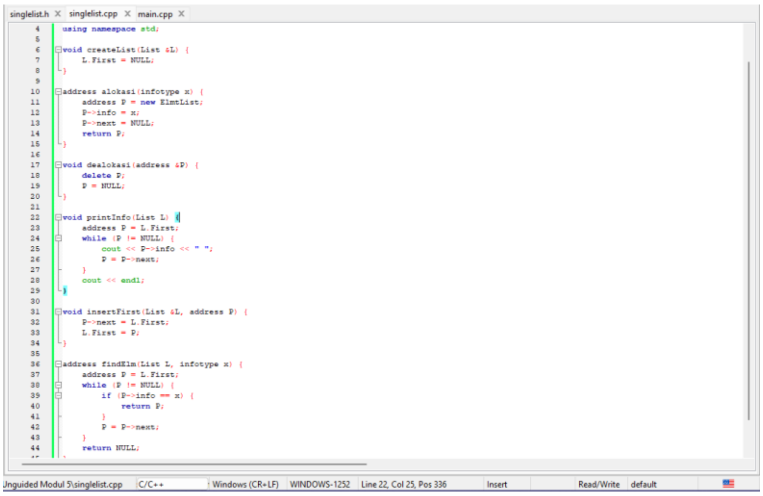
"C:\Users\aalfa\Documents\C++\Unguided Modul 5\Unguided Modul 5\bin\Debug\Unguided Modul 5.exe"

```
9 12 8 0 2

Process returned 0 (0x0)   execution time : 0.280 s
Press any key to continue.
```

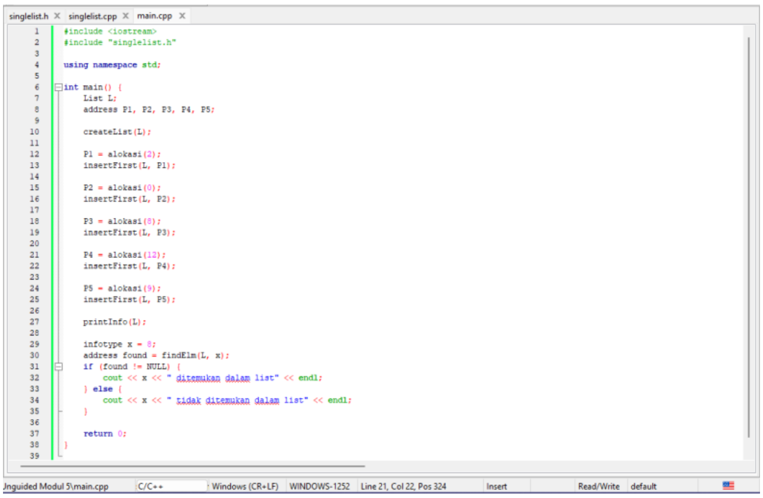
4.2. Kode Program :



```cpp
#ifndef SINGLELIST_H
#define SINGLELIST_H

// Deklarasi tipe data
typedef int infotype;
typedef struct ElmtList *address;

struct ElmtList {
    infotype info;
    address next;
};

struct List {
    address First;
};

// Deklarasi prosedur dan fungsi
void createList(List &L);
address alokasi(infotype x);
void dealokasi(address &P);
void printInfo(List L);
void insertFirst(List &L, address P);
address findElm(List L, infotype x);

#endif
```

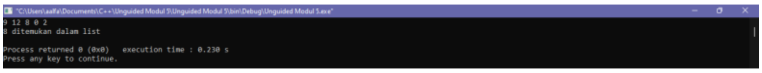


```cpp
using namespace std;

void createList(List &L) {
    L.First = NULL;
}

address alokasi(infotype x) {
    address P = new ElmtList;
    P->info = x;
    P->next = NULL;
    return P;
}

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

void printInfo(List L) {
    address P = L.First;
    while (P != NULL) {
        cout << P->info << " ";
        P = P->next;
    }
    cout << endl;
}

void insertFirst(List &L, address P) {
    P->next = L.First;
    L.First = P;
}

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



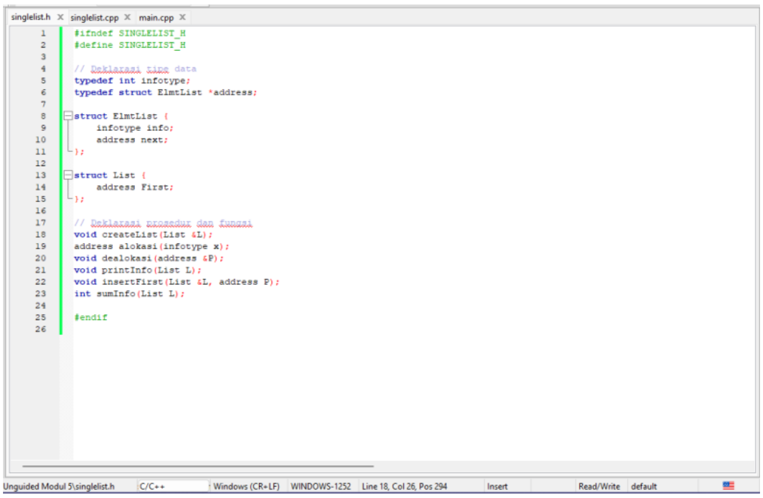
```cpp
#include <iostream>
#include "singlelist.h"

using namespace std;

int main() {
    List L;
    address P1, P2, P3, P4, P5;

    createList(L);

    P1 = alokasi(2);
    insertFirst(L, P1);

    P2 = alokasi(0);
    insertFirst(L, P2);

    P3 = alokasi(8);
    insertFirst(L, P3);

    P4 = alokasi(12);
    insertFirst(L, P4);

    P5 = alokasi(9);
    insertFirst(L, P5);

    printInfo(L);

    infotype x = 8;
    address found = findElm(L, x);
    if (found != NULL) {
        cout << x << " ditemukan dalam list" << endl;
    } else {
        cout << x << " tidak ditemukan dalam list" << endl;
    }

    return 0;
}
```

Ouput dari Kode Program :

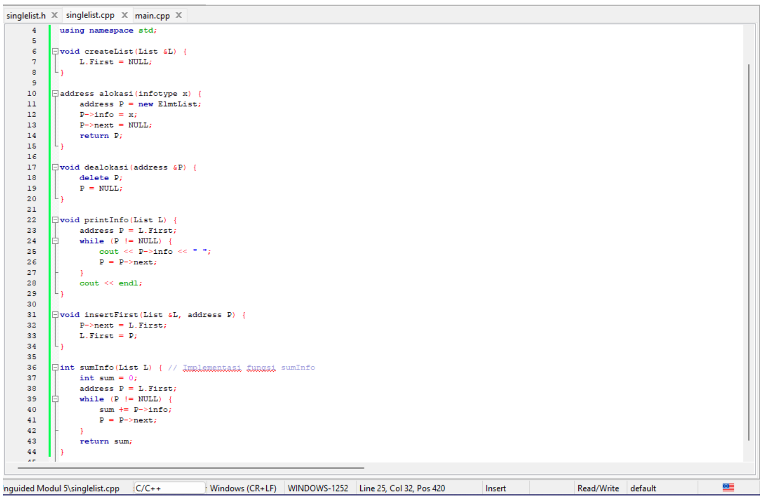

"C:\Users\aalfa\Documents\C++\Unguided Modul 5\Unguided Modul 5\bin\Debug\Unguided Modul 5.exe"

```
9 12 8 0 2
8 ditemukan dalam list

Process returned 0 (0x0)   execution time : 0.230 s
Press any key to continue.
```

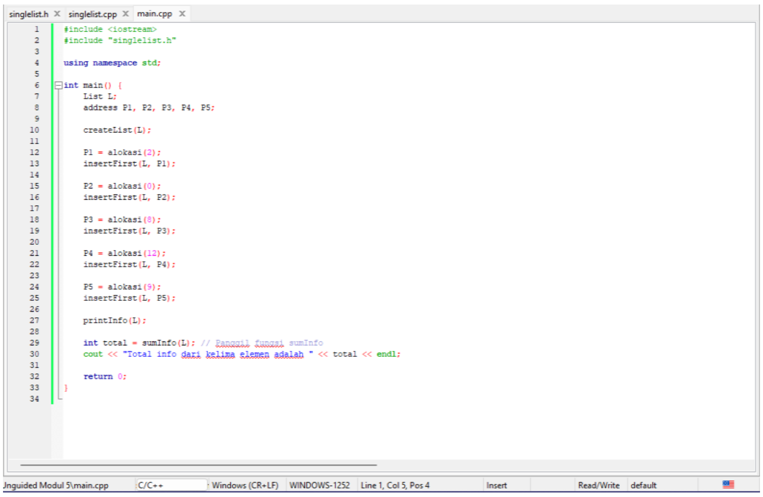
4.3. Kode Program :



```cpp
#ifndef SINGLELIST_H
#define SINGLELIST_H

// Deklarasi tipe data
typedef int infotype;
typedef struct ElmtList *address;

struct ElmtList {
    infotype info;
    address next;
};

struct List {
    address First;
};

// Deklarasi prosedur dan fungsi
void createList(List &L);
address alokasi(infotype x);
void dealokasi(address &P);
void printInfo(List L);
void insertFirst(List &L, address P);
int sumInfo(List L);

#endif
```

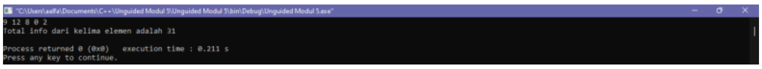


```cpp
using namespace std;

void createList(List &L) {
    L.First = NULL;
}

address alokasi(infotype x) {
    address P = new ElmtList;
    P->info = x;
    P->next = NULL;
    return P;
}

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

void printInfo(List L) {
    address P = L.First;
    while (P != NULL) {
        cout << P->info << " ";
        P = P->next;
    }
    cout << endl;
}

void insertFirst(List &L, address P) {
    P->next = L.First;
    L.First = P;
}

int sumInfo(List L) { // Implementasi fungsi sumInfo
    int sum = 0;
    address P = L.First;
    while (P != NULL) {
        sum += P->info;
        P = P->next;
    }
    return sum;
}
```



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

using namespace std;

int main() {
    List L;
    address P1, P2, P3, P4, P5;

    createList(L);

    P1 = alokasi(2);
    insertFirst(L, P1);

    P2 = alokasi(0);
    insertFirst(L, P2);

    P3 = alokasi(8);
    insertFirst(L, P3);

    P4 = alokasi(12);
    insertFirst(L, P4);

    P5 = alokasi(9);
    insertFirst(L, P5);

    printInfo(L);

    int total = sumInfo(L); // Panggil fungsi sumInfo
    cout << "Total info dari kelima elemen adalah " << total << endl;

    return 0;
}
```

Output dari Kode Program :


"C:\Users\aalfa\Documents\C++\Unguided Modul 5\Unguided Modul 5\bin\Debug\Unguided Modul 5.exe"

```
9 12 8 0 2
Total info dari kelima elemen adalah 31

Process returned 0 (0x0)   execution time : 0.211 s
Press any key to continue.
```