

**LAPORAN PRAKTIKUM**  
**MODUL 4**  
**“ Single Linked List 1”**



**Disusun Oleh:**  
**Kelvin Ferdinan - 2311104009**  
**S1SE07-01**

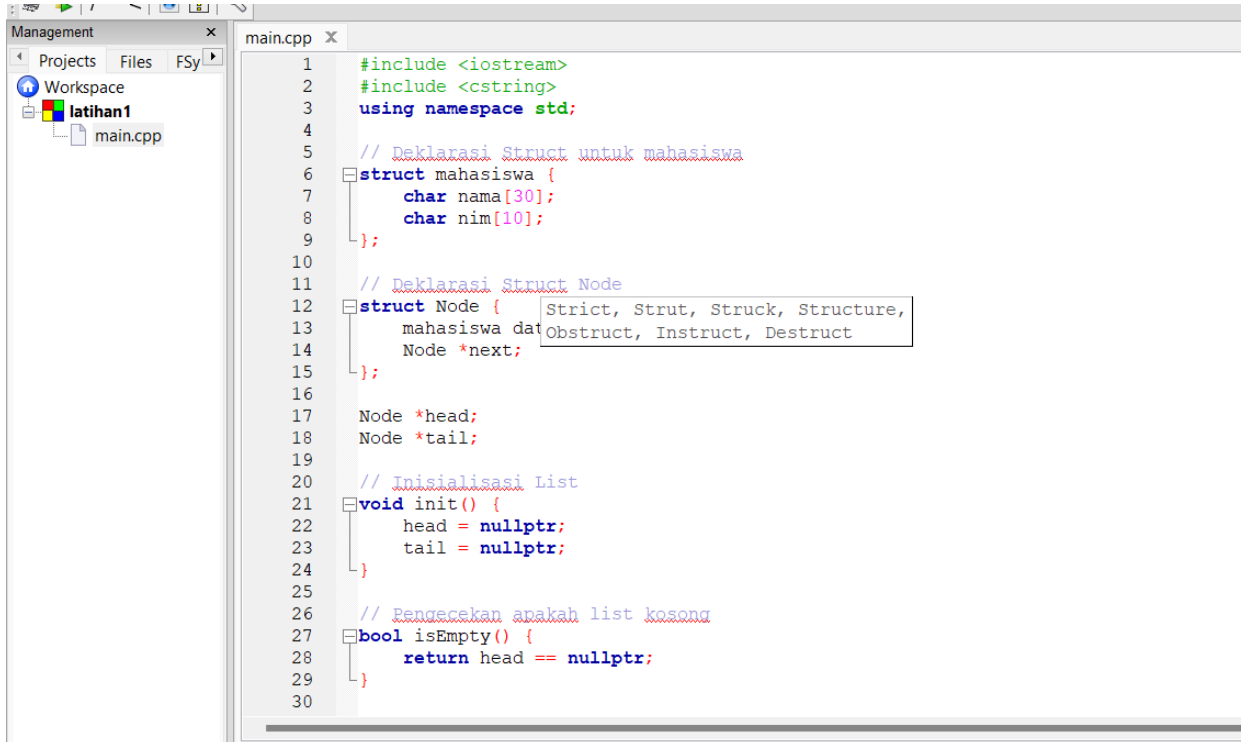
**Dosen :**  
**Yudha Islami Sulistya**

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

## A. Soal Tugas Pendahuluan

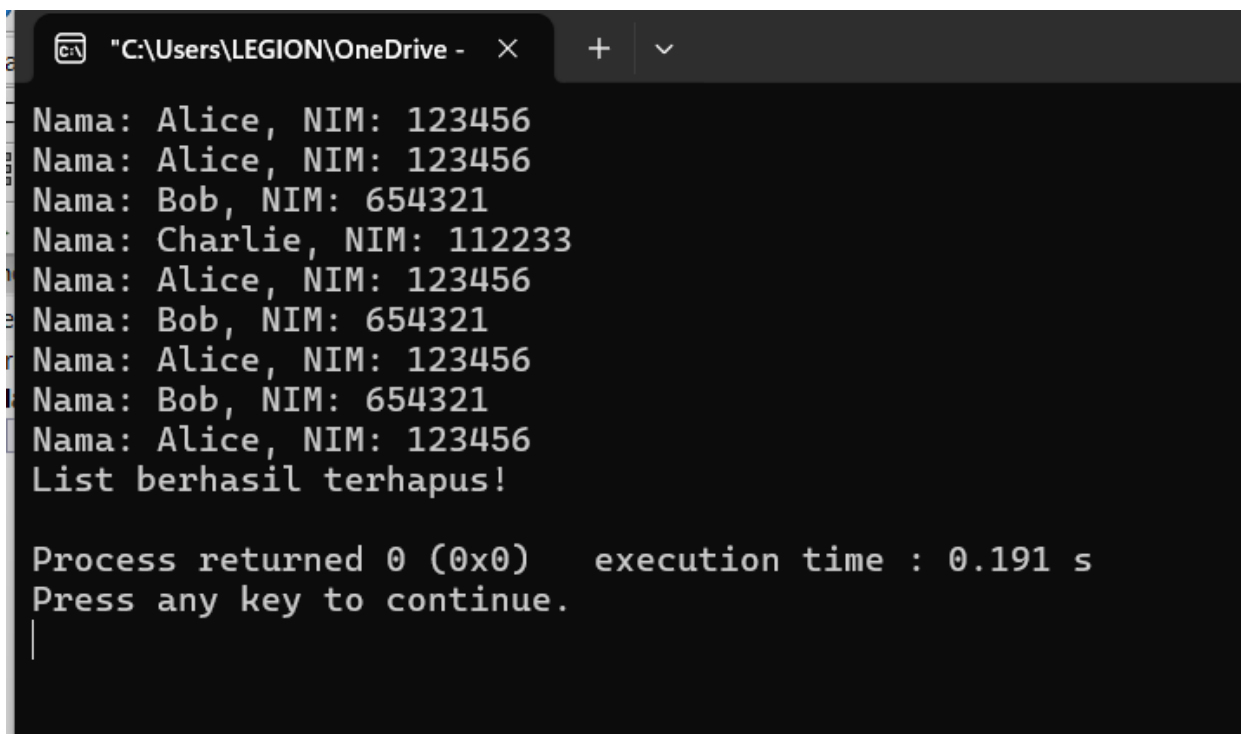
### I. Guided

1.



```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 // Deklarasi Struct untuk mahasiswa
6 struct mahasiswa {
7     char nama[30];
8     char nim[10];
9 };
10
11 // Deklarasi Struct Node
12 struct Node {
13     mahasiswa data;
14     Node *next;
15 };
16
17 Node *head;
18 Node *tail;
19
20 // Inisialisasi List
21 void init() {
22     head = nullptr;
23     tail = nullptr;
24 }
25
26 // Pengecekan apakah list kosong
27 bool isEmpty() {
28     return head == nullptr;
29 }
30
```

Hasil:



```
"C:\Users\LEGION\OneDrive - ...
Nama: Alice, NIM: 123456
Nama: Alice, NIM: 123456
Nama: Bob, NIM: 654321
Nama: Charlie, NIM: 112233
Nama: Alice, NIM: 123456
Nama: Bob, NIM: 654321
Nama: Alice, NIM: 123456
Nama: Bob, NIM: 654321
Nama: Alice, NIM: 123456
List berhasil terhapus!

Process returned 0 (0x0)   execution time : 0.191 s
Press any key to continue.
```

2.

```

1  #include <iostream>
2  using namespace std;
3
4  // Definisi struktur untuk elemen list
5  struct Node {
6      int data;           // Menyimpan nilai elemen
7      Node* next;        // Pointer ke elemen berikutnya
8  };
9
10 // Fungsi untuk mengalokasikan memori untuk node baru
11 Node* alokasi(int value) {
12     Node* newNode = new Node; // Alokasi memori untuk elemen baru
13     if (newNode != nullptr) { // Jika alokasi berhasil
14         newNode->data = value; // Mengisi data node
15         newNode->next = nullptr; // Set next ke nullptr
16     }
17     return newNode; // Mengembalikan pointer node baru
18 }
19
20 // Fungsi untuk dealokasi memori node
21 void dealokasi(Node* node) {
22     delete node; // Mengembalikan memori yang digunakan oleh node
23 }
24
25 // Pengecekan apakah list kosong
26 bool isEmpty(Node* head) {
27     return head == nullptr; // List kosong jika head adalah nullptr
28 }
29
30 // Menambahkan elemen di awal list

```

Hasil:

```

Isi List: 10 20 30
Jumlah elemen: 3
Isi List setelah penghapusan: List kosong!

Process returned 0 (0x0)   execution time : 0.059 s
Press any key to continue.

```

## II. Unguided

1.

```

1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int data;
6      Node* next;
7  };
8
9
10 void insertAtFront(Node*& head, int newData) {
11     Node* newNode = new Node();
12     newNode->data = newData;
13     newNode->next = head;
14     head = newNode;
15 }
16
17 void insertAtBack(Node*& head, int newData) {
18     Node* newNode = new Node();
19     newNode->data = newData;
20     newNode->next = nullptr;
21
22     if (head == nullptr) {
23         head = newNode;
24     } else {
25         Node* temp = head;
26         while (temp->next != nullptr) {
27             temp = temp->next;
28         }
29         temp->next = newNode;
30     }
31 }

```

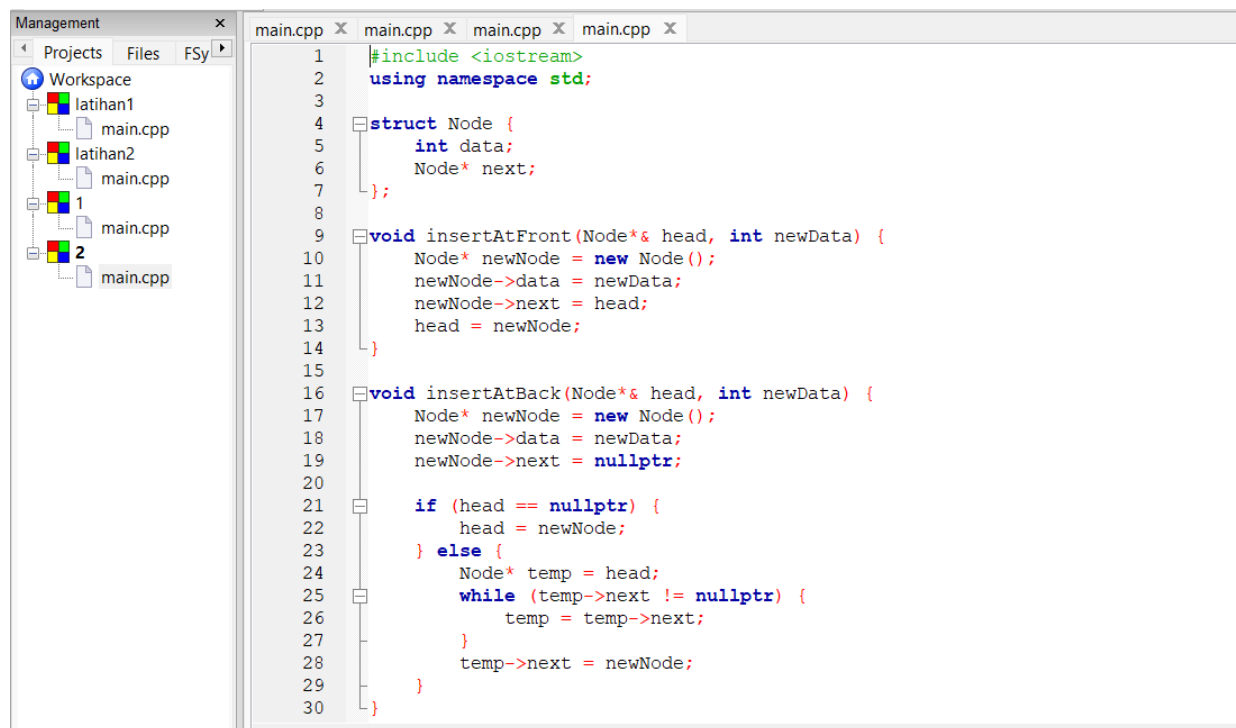
Hasil:

```
1 #include <iostream>

5 -> 10 -> 20

Process returned 0 (0x0)   execution time : 0.056 s
Press any key to continue.
```

2.



```
1 #include <iostream>
2 using namespace std;
3
4 struct Node {
5     int data;
6     Node* next;
7 };
8
9 void insertAtFront(Node*& head, int newData) {
10     Node* newNode = new Node();
11     newNode->data = newData;
12     newNode->next = head;
13     head = newNode;
14 }
15
16 void insertAtBack(Node*& head, int newData) {
17     Node* newNode = new Node();
18     newNode->data = newData;
19     newNode->next = nullptr;
20
21     if (head == nullptr) {
22         head = newNode;
23     } else {
24         Node* temp = head;
25         while (temp->next != nullptr) {
26             temp = temp->next;
27         }
28         temp->next = newNode;
29     }
30 }
```

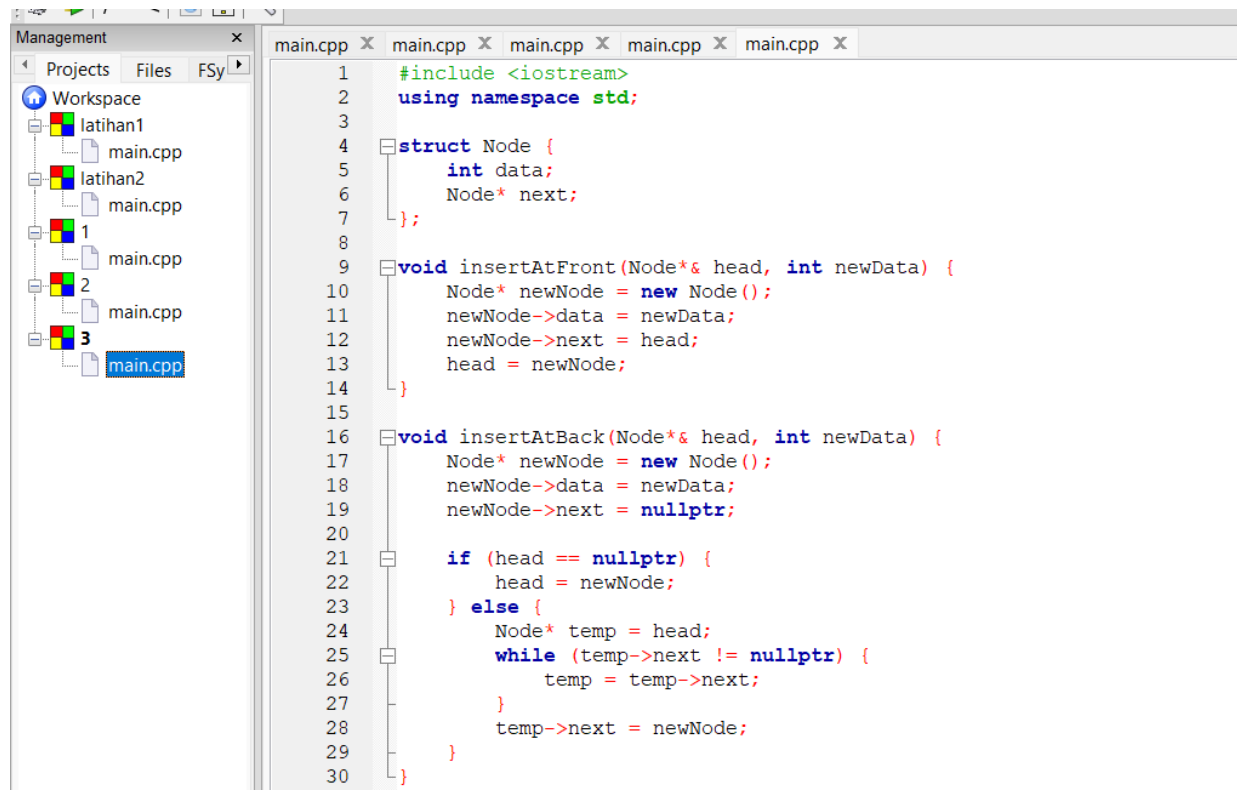
Hasil:

```
"C:\Users\LEGION\OneDrive - X + v

linked list awal: 5 -> 10 -> 20
setelah dihapus: 5 -> 20

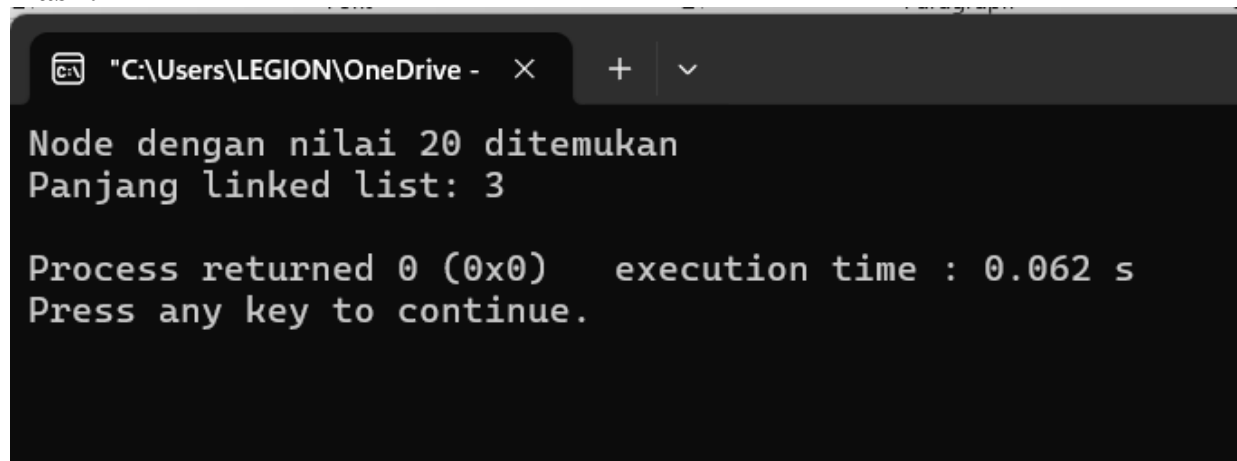
Process returned 0 (0x0)   execution time : 0.073 s
Press any key to continue.
```

3.



```
1 #include <iostream>
2 using namespace std;
3
4 struct Node {
5     int data;
6     Node* next;
7 };
8
9 void insertAtFront(Node*& head, int newData) {
10     Node* newNode = new Node();
11     newNode->data = newData;
12     newNode->next = head;
13     head = newNode;
14 }
15
16 void insertAtBack(Node*& head, int newData) {
17     Node* newNode = new Node();
18     newNode->data = newData;
19     newNode->next = nullptr;
20
21     if (head == nullptr) {
22         head = newNode;
23     } else {
24         Node* temp = head;
25         while (temp->next != nullptr) {
26             temp = temp->next;
27         }
28         temp->next = newNode;
29     }
30 }
```

Hasil:

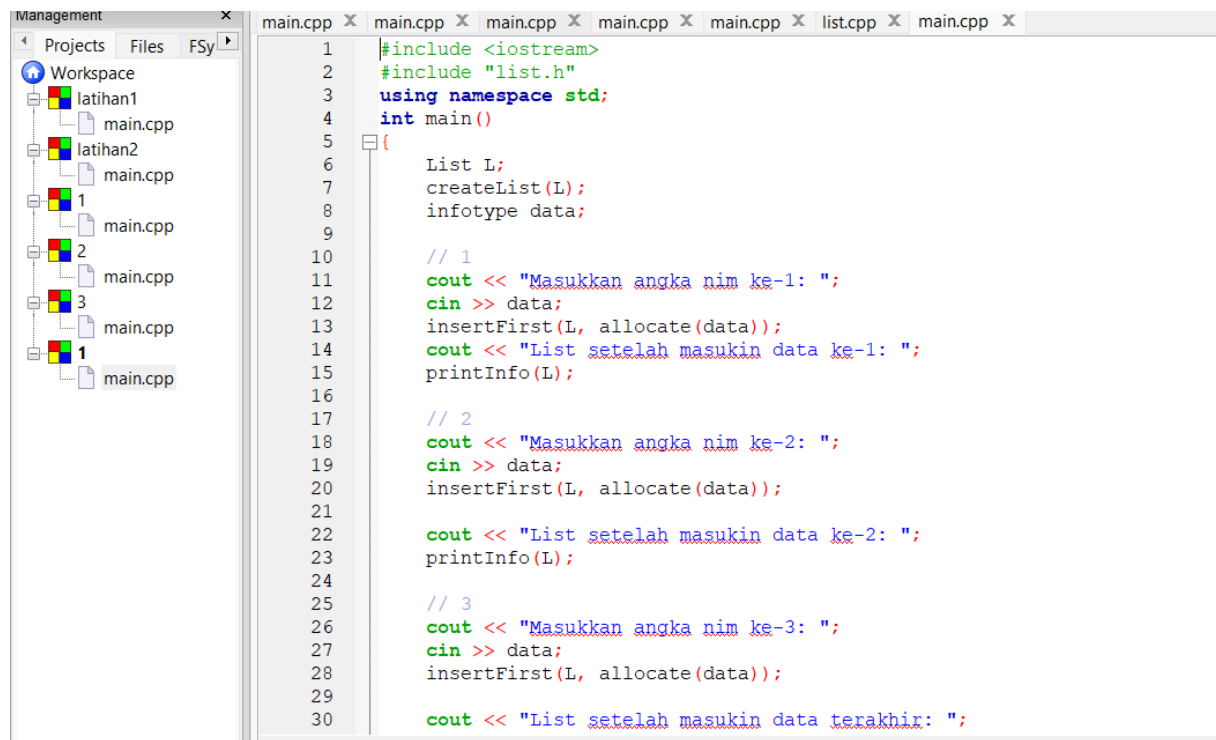


```
"C:\Users\LEGION\OneDrive - X + v
Node dengan nilai 20 ditemukan
Panjang linked list: 3

Process returned 0 (0x0)   execution time : 0.062 s
Press any key to continue.
```

### III. TP

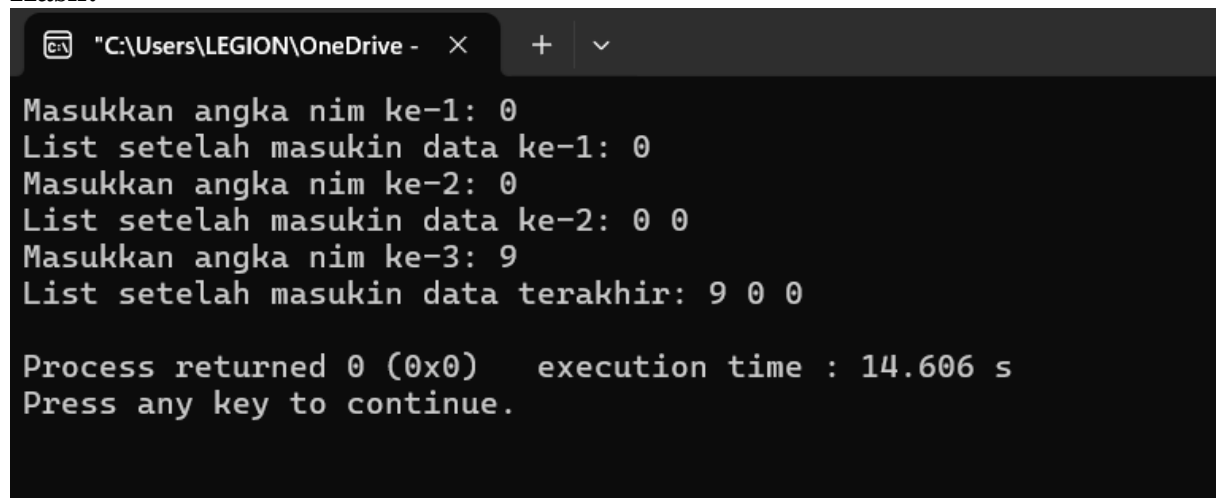
1.



The image shows a C++ IDE with a project structure on the left and a code editor on the right. The project structure includes a 'Workspace' folder with subfolders 'latihan1' and 'latihan2', each containing a 'main.cpp' file. The code editor displays the contents of 'main.cpp', which includes headers, namespace declarations, and a main function that interacts with a 'List' object.

```
1 #include <iostream>
2 #include "list.h"
3 using namespace std;
4 int main()
5 {
6     List L;
7     createList(L);
8     infotype data;
9
10    // 1
11    cout << "Masukkan angka nim ke-1: ";
12    cin >> data;
13    insertFirst(L, allocate(data));
14    cout << "List setelah masukin data ke-1: ";
15    printInfo(L);
16
17    // 2
18    cout << "Masukkan angka nim ke-2: ";
19    cin >> data;
20    insertFirst(L, allocate(data));
21
22    cout << "List setelah masukin data ke-2: ";
23    printInfo(L);
24
25    // 3
26    cout << "Masukkan angka nim ke-3: ";
27    cin >> data;
28    insertFirst(L, allocate(data));
29
30    cout << "List setelah masukin data terakhir: ";
```

Hasil:



The image shows a terminal window with the output of the C++ program. The program prompts the user to enter three numbers (nim) and displays the resulting list after each insertion. The execution time is 14.606 seconds.

```
"C:\Users\LEGION\OneDrive - x + v
Masukkan angka nim ke-1: 0
List setelah masukin data ke-1: 0
Masukkan angka nim ke-2: 0
List setelah masukin data ke-2: 0 0
Masukkan angka nim ke-3: 9
List setelah masukin data terakhir: 9 0 0

Process returned 0 (0x0)   execution time : 14.606 s
Press any key to continue.
```

Semoga Selalu diberi kemudahan^^