test.cpp

```cpp
#include <iostream>
using namespace std;

struct Node {
    int noMhs;
    string name;
    Node* next;
};

Node* START = NULL;

void addNode() {
    int nim;
    string nama;
    Node* nodeBaru = new Node();
    cout << "Masukkan NIM: ";
    cin >> nim;
    cout << "Masukkan Nama: ";
    cin >> nama;
    nodeBaru->noMhs = nim;
    nodeBaru->name = nama;

    if (START == NULL || nim <= START->noMhs) {
        if (START != NULL && nim == START->noMhs)
        {
            cout << "NIM sudah ada" << endl;
            return;
        }

        nodeBaru->next = START;
        START = nodeBaru;
        return;
    }

    Node* previous = START;
    Node* current = START;

    while ((current != NULL) && (nim >= current->noMhs))
    {
        if (nim == current->noMhs)
        {
            cout << "NIM sudah ada" << endl;
            return;
        }
        previous = current;
        current = current->next;
    }

    nodeBaru->next = current;
    previous->next = nodeBaru;
}

```

```
53  bool serachNode(int nim, Node* current, Node* previous) {
54      previous = START;
55      current = START;
56      while (current != NULL && nim > current->noMhs)
57      {
58          previous = current;
59          current = current->next;
60      }
61
62      if (current == NULL)
63      {
64          return false;
65      }
66      else if (current->noMhs == nim)
67      {
68          return true;
69      }
70      else
71      {
72          return false;
73      }
74  }
75  bool deleteNode(int nim) {
76      Node* current = START;
77      Node* previous = START;
78      if (serachNode(nim, previous, current) == false)
79          return false;
80      previous->next = current->next;
81      if (current == START)
82          START = current->next;
83      return true;
84  }
85  bool listEmpty() {
86      if (START == NULL)
87          return true;
88      else
89          return false;
90
91  }
92
93  void traverse() {
94      if (listEmpty()) {
95          cout << "List Kosong" << endl;
96          system("pause");
97          system("cls");
98          return;
99      }
100     else {
101         Node* currentNode = START;
102         while (currentNode != NULL) {
103             cout << "NIM: " << currentNode->noMhs << ", Nama: " << currentNode->name << endl;
104             currentNode = currentNode->next;
105         }
106     }
107 }
108
```

```
void searchData() {
    if (listEmpty()) {
        cout << "List Kosong" << endl;
        system("pause");
        system("cls");
        return;
    }
    else {
        int nim;
        cout << "Masukkan NIM: ";
        cin >> nim;
        Node* currentNode = START;
        while (currentNode != NULL) {
            if (currentNode->noMhs == nim) {
                cout << "NIM: " << currentNode->noMhs << ", Nama: " << currentNode->name << endl;
                return;
            }
            currentNode = currentNode->next;
        }
        cout << "Data tidak ditemukan" << endl;
    }
}

int main() {
    int pilihan;
    do
    {
        try
        {

            cout << "1. Tambah Data" << endl;
            cout << "2. Hapus Data" << endl;
            cout << "3. Tampilkan Data" << endl;
            cout << "4. Cari Data" << endl;
            cout << "5. Keluar" << endl;
            cout << "Pilihan: ";
            cin >> pilihan;
            switch (pilihan)
            {
            case 1:
                addNode();
                cout << "Data Berhasil Ditambahkan" << endl;
                system("pause");
                system("cls");
                break;
            case 2:
                if (listEmpty())
                {
                    cout << "List kosong" << endl;
                    system("pause");
                    system("cls");
                    break;
                }

                int nim;
```

```
                cout << "Masukkan NIM: ";
                cin >> nim;
                if (deleteNode(nim)) {
                    cout << "nim: " << nim << " berhasil dihapus" << endl;
                    system("pause");
                    system("cls");
                }
                else
                    cout << "Data tidak ditemukan" << endl;
                break;
            case 3:
                traverse();
                break;
            case 4:
                searchData();
                break;
            case 5:
                break;
            default:
                cout << "Pilihan tidak ada" << endl;
                break;
            }
        }
        catch (exception e)
        {
            cout << "Terjadi kesalahan" << endl;
        }

    } while (pilihan != 5);
}

```