

test.cpp

```
1  #include <iostream>
2  using namespace std;
3
4  struct Node {
5      int noMhs;
6      string name;
7      Node* next;
8  };
9
10 Node* START = NULL;
11
12 void addNode() {
13     int nim;
14     string nama;
15     Node* nodeBaru = new Node();
16     cout << "Masukkan NIM: ";
17     cin >> nim;
18     cout << "Masukkan Nama: ";
19     cin >> nama;
20     nodeBaru->noMhs = nim;
21     nodeBaru->name = nama;
22
23     if (START == NULL || nim <= START->noMhs) {
24         if (START != NULL && nim == START->noMhs)
25         {
26             cout << "NIM sudah ada" << endl;
27             return;
28         }
29
30         nodeBaru->next = START;
31         START = nodeBaru;
32         return;
33     }
34
35     Node* previous = START;
36     Node* current = START;
37
38     while ((current != NULL) && (nim >= current->noMhs))
39     {
40         if (nim == current->noMhs)
41         {
42             cout << "NIM sudah ada" << endl;
43             return;
44         }
45         previous = current;
46         current = current->next;
47     }
48
49     nodeBaru->next = current;
50     previous->next = nodeBaru;
51 }
52
```

```

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

```

```

109 void searchData() {
110     if (listEmpty()) {
111         cout << "List Kosong" << endl;
112         system("pause");
113         system("cls");
114         return;
115     }
116     else {
117         int nim;
118         cout << "Masukkan NIM: ";
119         cin >> nim;
120         Node* currentNode = START;
121         while (currentNode != NULL) {
122             if (currentNode->noMhs == nim) {
123                 cout << "NIM: " << currentNode->noMhs << ", Nama: " << currentNode->name <<
endl;
124                 return;
125             }
126             currentNode = currentNode->next;
127         }
128         cout << "Data tidak ditemukan" << endl;
129     }
130 }
131
132 int main() {
133     int pilihan;
134     do
135     {
136         try
137         {
138
139             cout << "1. Tambah Data" << endl;
140             cout << "2. Hapus Data" << endl;
141             cout << "3. Tampilkan Data" << endl;
142             cout << "4. Cari Data" << endl;
143             cout << "5. Keluar" << endl;
144             cout << "Pilihan: ";
145             cin >> pilihan;
146             switch (pilihan)
147             {
148                 case 1:
149                     addNode();
150                     cout << "Data Berhasil Ditambahkan" << endl;
151                     system("pause");
152                     system("cls");
153                     break;
154                 case 2:
155                     if (listEmpty())
156                     {
157                         cout << "List kosong" << endl;
158                         system("pause");
159                         system("cls");
160                         break;
161                     }
162
163                     int nim;

```

```

164         cout << "Masukkan NIM: ";
165         cin >> nim;
166         if (deleteNode(nim)) {
167             cout << "nim: " << nim << " berhasil dihapus" << endl;
168             system("pause");
169             system("cls");
170         }
171         else
172             cout << "Data tidak ditemukan" << endl;
173         break;
174     case 3:
175         traverse();
176         break;
177     case 4:
178         searchData();
179         break;
180     case 5:
181         break;
182     default:
183         cout << "Pilihan tidak ada" << endl;
184         break;
185     }
186 }
187 catch (exception e)
188 {
189     cout << "Terjadi kesalahan" << endl;
190 }
191
192 } while (pilihan != 5);
193 }
194

```