

```

1 #include <iostream>
2 #include <string>
3 #include <set>
4
5 using namespace std;
6
7 // Struktur data Mahasiswa
8 struct Mahasiswa {
9     string nama;
10    string NIM;
11    string kelas;
12    int nilaiAsesmen;
13    int nilaiPraktikum;
14 };
15
16 // Struktur Node untuk Double Linked List
17 struct Node {
18     Mahasiswa data;
19     Node* next;
20     Node* prev;
21 };
22
23 // Struktur list
24 struct List {
25     Node* first;
26     Node* last;
27 };
28
29 // Function untuk membuat elemen baru
30 Node* newElement(Mahasiswa data) {
31     Node* newNode = new Node;
32     newNode->data = data;
33     newNode->next = nullptr;
34     newNode->prev = nullptr;
35     return newNode;
36 }
37
38 // Function untuk membuat list baru
39 List* createNewList() {
40     List* list;
41     list->first = nullptr;
42     list->last = nullptr;
43     return list;
44 }
45
46 // Procedure untuk menambahkan elemen di awal list
47 void insertFirst(List &list, Node* p) {
48     if (list->first == nullptr) {
49         list->first = list->last = p;
50     } else {
51         p->next = list->first;
52         list->first->prev = p;
53         list->first = p;
54     }
55 }
56
57 // Procedure untuk menambahkan elemen di akhir list
58 void insertLast(List &list, Node* p) {
59     if (list->last == nullptr) {
60         list->first = list->last = p;
61     } else {
62         p->prev = list->last;
63         list->last->next = p;
64         list->last = p;
65     }
66 }
67
68 // Procedure untuk menambahkan data mahasiswa
69 void addData(List &list, Mahasiswa data) {
70     Node* p = newElement(data);
71     if (stoi(data.NIM) % 2 == 0) { // NIM genap
72         insertLast(list, p);
73     } else { // NIM ganjil
74         insertFirst(list, p);
75     }
76 }
77
78 // Procedure untuk mencetak isi list
79 void printList(List list) {
80     if (list->first == nullptr) {
81         cout << "List kosong." << endl;
82         return;
83     }
84     Node* current = list->first;
85     while (current != nullptr) {
86         cout << "Nama: " << current->data.nama
87              << ", NIM: " << current->data.NIM
88              << ", Kelas: " << current->data.kelas
89              << ", Nilai Asesmen: " << current->data.nilaiAsesmen
90              << ", Nilai Praktikum: " << current->data.nilaiPraktikum << endl;
91         current = current->next;
92     }
93 }
94
95 // Function untuk mencari mahasiswa dengan nilai asesmen tertinggi
96 void findMaxAsesmen(List list) {
97     Node* current = list->first;
98     Node* maxNode = nullptr;
99     int maxValue = -1;
100
101     while (current != nullptr) {
102         if (current->data.nilaiAsesmen > maxValue) {
103             maxValue = current->data.nilaiAsesmen;
104             maxNode = current;
105         }
106         current = current->next;
107     }
108
109     if (maxNode != nullptr) {
110         cout << "Mahasiswa dengan nilai asesmen tertinggi: " << endl;
111         cout << "Nama: " << maxNode->data.nama
112              << ", NIM: " << maxNode->data.NIM
113              << ", Nilai: " << maxNode->data.nilaiAsesmen << endl;
114     } else {
115         cout << "List kosong, tidak ada data." << endl;
116     }
117 }
118
119 // Procedure untuk menghapus data duplikat berdasarkan NIM
120 void removeDuplicates(List &list) {
121     Node* current = list->first;
122
123     while (current != nullptr) {
124         Node* check = current->next;
125         while (check != nullptr) {
126             if (current->data.NIM == check->data.NIM) {
127                 Node* temp = check;
128                 check = check->next;
129                 // Update pointer untuk node sebelum dan sesudah
130                 if (temp->prev) temp->prev->next = temp->next;
131                 if (temp->next) temp->next->prev = temp->prev;
132                 // Perbarui pointer first/last jika elemen dihapus
133                 if (temp == list->first) list->first = temp->next;
134                 if (temp == list->last) list->last = temp->prev;
135                 delete temp;
136             } else {
137                 check = check->next;
138             }
139             current = current->next;
140         }
141     }
142 }
143
144 // Main Program
145 int main() {
146     List list = createNewList();
147     int N;
148
149     // Tambahkan data
150     cout << "Masukkan jumlah data mahasiswa yang akan ditambahkan: ";
151     cin >> N;
152
153     for (int i = 0; i < N; i++) {
154         Mahasiswa m;
155         cout << "Masukkan data mahasiswa ke-" << i + 1 << endl;
156         cout << "Nama: ";
157         cin >> m.nama;
158         cout << "NIM: ";
159         cin >> m.NIM;
160         cout << "Kelas: ";
161         cin >> m.kelas;
162         cout << "Nilai Asesmen: ";
163         cin >> m.nilaiAsesmen;
164         cout << "Nilai Praktikum: ";
165         cin >> m.nilaiPraktikum;
166         addData(list, m);
167     }
168
169     // Cetak data
170     cout << "\nData Mahasiswa: " << endl;
171     printList(list);
172
173     // Cari nilai asesmen tertinggi
174     cout << "\nMencari nilai asesmen tertinggi..." << endl;
175     findMaxAsesmen(list);
176
177     // Hapus duplikat
178     cout << "\nMenghapus data duplikat..." << endl;
179     removeDuplicates(list);
180
181     // Cetak data setelah penghapusan duplikat
182     cout << "\nData Mahasiswa Setelah Penghapusan Duplikat: " << endl;
183     printList(list);
184
185     return 0;
186 }

```

```
Masukkan jumlah data mahasiswa yang akan ditambahkan: 3
Masukkan data mahasiswa ke-1
Nama: Haza
NIM: 12345
Kelas: B
Nilai Asesmen: 90
Nilai Praktikum: 90
Masukkan data mahasiswa ke-2
Nama: Zaidan
NIM: 23546
Kelas: A
Nilai Asesmen: 70
Nilai Praktikum: 70
Masukkan data mahasiswa ke-3
Nama: Zidna
NIM: 45670
Kelas: B
Nilai Asesmen: 80
Nilai Praktikum: 80

Data Mahasiswa:
Nama: Haza, NIM: 12345, Kelas: B, Nilai Asesmen: 90, Nilai Praktikum: 90
Nama: Zaidan, NIM: 23546, Kelas: A, Nilai Asesmen: 70, Nilai Praktikum: 70
Nama: Zidna, NIM: 45670, Kelas: B, Nilai Asesmen: 80, Nilai Praktikum: 80

Mencari nilai asesmen tertinggi...
Mahasiswa dengan nilai asesmen tertinggi:
Nama: Haza, NIM: 12345, Nilai: 90

Menghapus data duplikat...

Data Mahasiswa Setelah Penghapusan Duplikat:
Nama: Haza, NIM: 12345, Kelas: B, Nilai Asesmen: 90, Nilai Praktikum: 90
Nama: Zaidan, NIM: 23546, Kelas: A, Nilai Asesmen: 70, Nilai Praktikum: 70
Nama: Zidna, NIM: 45670, Kelas: B, Nilai Asesmen: 80, Nilai Praktikum: 80
```